4600 Series IP Telephone

LAN Administrator's Guide

555-233-507
Issue 1.7
July 2002

Copyright 2002, Avaya Inc.
All Rights Reserved, Printed in U.S.A.

Notice
Every effort was made to ensure that the information in this book was complete and accurate at the time of printing. However, information is subject to change.

Avaya Web Page
The world wide web home page for Avaya is:
http://www.avaya.com

Preventing Toll Fraud
Toll fraud is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or working on your company's behalf). Be aware that there is a risk of toll fraud associated with your system and that, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Fraud Intervention
If you suspect that you are being victimized by toll fraud and you need technical support or assistance and are in within the United States, call the Technical Service Center Toll Fraud Intervention Hotline at 1.800.643.2353. If you need technical support or assistance and are outside of the United States, contact the equipment vendor from whom you purchased your equipment service maintenance contract. If you need to report toll fraud issues regarding a public telephone, contact the in-country telephone service provider.

Providing Telecommunications Security
Telecommunications security of voice, data, and/or video communications is the prevention of any type of intrusion to, that is, either unauthorized or malicious access to or use of, your company's telecommunications equipment by some party.

Your company's "telecommunications equipment" includes both this Avaya product and any other voice/data/video equipment that could be accessed via this Avaya product (that is, "networked equipment").

An "outside party" is anyone who is not a corporate employee, agent, subcontractor, or working on your company's behalf. Whereas, a "malicious party" is anyone, including someone who may be otherwise authorized, who accesses your telecommunications equipment with either malicious or mischievous intent.

Such intrusions may be either to/through synchronous (time-multiplexed and/or circuit-based) or asynchronous (character-, message-, or packet-based) equipment or interfaces for reasons of:

- Utilization (of capabilities special to the accessed equipment)
- Theft (such as, of intellectual property, financial assets, or toll-facility access)
- Eavesdropping (privacy invasions to humans)
- Mischief (troubling, but apparently innocuous, tampering)
- Harm (such as harmful tampering, data loss or alteration, regardless of motive or intent)

Be aware that there may be a risk of unauthorized intrusions associated with your system and/or its networked equipment. Also realize that, if such an intrusion should occur, it could result in a variety of losses to your company, including but not limited to, human/data privacy, intellectual property, material assets, financial resources, labor costs, and/or legal costs.

Federal Communications Commission Statement
Part 68: Network Registration Number. This equipment is registered with the FCC in accordance with Part 68 of the FCC Rules. It is identified by FCC registration number AV1USA-43058-MF-E.

Part 68: Answer-Supervision Signaling. Allowing this equipment to be operated in a manner that does not provide proper answer-supervision signaling is in violation of Part 68 Rules. This equipment returns answer-supervision signals to the public switched network when:

- Answered by the called station
- Answered by the attendant
- Routed to a recorded announcement that can be administered by the CPE user

This equipment returns answer-supervision signals on all DID calls forwarded back to the public switched telephone network. Permissible exceptions are:

- A call is unanswered
- A busy tone is received
- A reorder tone is received

Industry Canada (IC) Interference Information
NOTICE: This equipment meets the applicable Industry Canada Terminal Equipment Technical Specifications. This is confirmed by the registration number. The abbreviation, IC, before the registration number signifies that registration was performed based on a Declaration of Conformity indicating that Industry Canada technical specifications were met. It does not imply that Industry Canada approved the equipment.

Le Présent Appareil Nomérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la class A préscrites dans le reglement sur le brouillage radioélectrique édicté par le Industrie Canada.

Trademarks
DEFINITY is a registered trademark of Avaya, Inc.
MultiVantage is a trademark of Avaya, Inc.

Ordering Information
Call: Avaya Publications Center
U.S. and Canada Voice 1 800 457 1235
Outside U.S. and Canada Voice +1 207 866 6701
U.S. and Canada Fax 1 800 457 1764
Outside U.S. and Canada Fax +1 207 626 7269
Write: GlobalWare Solutions
200 Ward Hill Avenue
Haverhill, MA 01835 USA
Attention: Avaya Account Manager
Email: totalware@gwsmail.com

For additional documents, refer to the section in About This Guide titled Related Documents. An online copy of this and other related Avaya product documentation can be found at: http://www.avaya.com/support.

Obtaining Products

To learn more about Avaya products and to order products, access the Avaya web site at http://www.avaya.com. Or call the following numbers: customers 1 800 451 2100, account executives 1 888 778 1880 (voice) or 1 888 778 1881 (fax).

European Union Declaration of Conformity

The “CE” mark affixed to the equipment means that it conforms to the referenced European Union (EU) Directives listed below:

EMC Directive	89/336/EEC
Low-Voltage Directive	73/23/EEC

For more information on standards compliance, contact your local distributor.

4600 Series IP Telephone LAN Administrator's Guide

Table of Contents

1 Introduction

About This Guide 1-1

Intended Audience 1-1

Document Organization 1-2
- Change History 1-2
- Terms Used in This Guide 1-3
- Conventions Used in This Guide 1-4

Online Documentation 1-5

Related Documents 1-5

Customer Support 1-8

2 Overview of Voice over IP (VoIP)

Introduction 2-1

Overview of Voice over IP 2-1
- Data and Voice Network Similarities 2-1
- Delay and Jitter 2-2
- Tandem Coding 2-2
- Voice Coding Standards 2-3
- H.323 Standard 2-3
- DHCP 2-3
- TFTP 2-3
- NAT 2-4
- QoS 2-4

SNMP 2-4

Network Assessment 2-5

Suggestions for Installation and Configuration 2-5
- Reliability and Performance 2-5
- IP Address Lists and Station Number Portability 2-6
- Security 2-6

4600 Series IP Telephones 2-7
- Dual Connection Architecture 2-7
- Single Connection Architecture 2-7
- Registration and Authentication 2-7

- Software 2-7
- WAN Considerations 2-7
- DHCP and TFTP Servers 2-7

Initialization Process 2-8
- Step 1: Telephone to Network 2-8
- Step 2: DHCP Server to Telephone 2-8
- Step 3: Telephone and TFTP Server 2-9
- Step 4: Telephone and the DEFINITY®/MultiVantage*TM* Call Server 2-9

3 Requirements

Introduction 3-1

Hardware Requirements 3-1

Software Requirements 3-3

4 Server Administration

Introduction 4-1

Administering 4600 Series IP Telephones on DEFINITY®/MultiVantage™ Servers 4-4
- DEFINITY® Release 8.4. 4-4
- DEFINITY® Releases 9, 9.5, 10, and MultiVantage*TM* Release 11 4-4

DHCP and TFTP 4-4

Software Checklist 4-5

Required Network Information 4-5

DHCP 4-6
- Choosing a DHCP Configuration 4-6
- DHCP Software Alternatives. 4-6
- DHCP Generic Setup 4-7
- Windows NT 4.0 DHCP Server. 4-9
- Windows 2000 DHCP Server 4-13

TFTP 4-17
- TFTP Generic Setup. 4-17
- Avaya TFTP (Suite Pro) 4-17

4600 Series IP Telephone Scripts and Application Files 4-18
- Contents of the Upgrade Script 4-19

QoS 4-20
- IEEE 802.1D and 802.1Q 4-20
- DIFFSERV 4-21
- UDP Port Selection. 4-22
- QoS with 4620 and 4630 IP Telephones 4-22
- RSVP and RTCP 4-23

Administering Options for the 4600 Series IP Telephones 4-24
- DNS Addressing 4-24
- Customizing the Site-Specific Option Number (SSON) 4-25
- Entering Options via the Telephone Dialpad 4-25
- Customizing the 4630 IP Telephone 4-26
- 4630 Backup/Restore 4-30
- Call Log Archive 4-32
- Customizing the 4620 IP Telephone 4-32

5 Troubleshooting Guidelines

Introduction 5-1

Error Conditions 5-1

The View Administrative Option 5-8

Error Messages 5-9

Appendix A:Avaya - 46xx IP Telephone MIB

Avaya - 46xx IP Telephone MIB A-1

Appendix B:Creating Websites for the 4630 IP Telephone

Introduction B-1

General Background B-1

Browser Features and Behavior B-2
- Document Skeleton B-2
- Content-Based Style B-3
- Logical Style B-4
- Physical Style B-4
- Physical Spacing and Layout B-5
- Lists and Tables B-5
- Images B-7
- Links B-7
- Frames B-8
- Forms B-8
- Character Entities B-9
- Colors B-9
- Fonts B-9
- Cookies B-10

Design Guidelines B-10
- Fixed-Width Objects B-10
- Images B-11
- Frames B-11
- Fonts B-11
- Maintaining Context B-12
- User Interaction B-12

Appendix C:Creating Websites for the 4620 IP Telephone

Introduction C-1
General Background C-1
WML Document Skeleton C-2
Text Elements C-4
Text Formatting Tags C-5
Anchor Elements C-5
Image Elements C-6
Event Elements C-7
Task Elements C-10
Input Elements C-11
Variable Elements C-14
Character Entities C-15
Colors and Fonts C-16
Summary Of WML Tags And Attributes C-16

Introduction

1

About This Guide

This guide provides a description of Voice over IP, describes how to administer the DHCP and TFTP servers and covers how to troubleshoot operational problems with the 4600 Series IP Telephones and the servers.

The 4600 Series IP Telephone product line is a supplement to Avaya's IP Solutions platform.

NOTE Unless otherwise indicated, references in this document to "the DEFINITY® server" also refer to the MultiVantage™ media servers.

Intended Audience

This document is intended for personnel administering the DHCP and TFTP servers to support the 4600 Series IP Telephones and those administering the Local Area Network (LAN) itself.

⚠ CAUTION:

Many of the products mentioned in this document are not supported by Avaya. Care should be taken to ensure there is adequate technical support available for the TFTP and DHCP servers. If the TFTP or DHCP servers are not functioning correctly, the 4600 Series IP Telephones may not operate correctly.

Document Organization

The guide contains the following sections:

Chapter 1, Introduction	Provides an overview of the 4600 Series IP Telephone Administrator's document.
Chapter 2, Overview of Voice over IP (VoIP)	Describes VoIP and factors influencing its performance that must be considered when implementing this feature.
Chapter 3, Requirements	Describes the hardware and software requirements for Avaya's VoIP offering.
Chapter 4, Server Administration	Describes the administration of DHCP and TFTP for the 4600 Series IP Telephones.
Chapter 5, Troubleshooting Guidelines	Describes messages that may occur during the operation of the 4600 Series IP Telephones.
Appendix A, Avaya - 46xx IP Telephone MIB	Provides the MIB specification for the 46xx IP Telephones (4606, 4612, 4624, and 4630).
Appendix B, Creating Websites for the 4630 IP Telephone	Provides information on creating and customizing websites for viewing on the 4630 IP Telephone. Also describes the current capabilities and limitations of the 4630's web browser.
Appendix C, Creating Websites for the 4620 IP Telephone	Provides information on creating and customizing websites for viewing on the 4620 IP Telephone. Also describes the current capabilities and limitations of the 4620's web browser.

Change History

Issue 1.0	This document was issued for the first time in November 2000.
Issue 1.1	This version of the document, revised and issued in April 2001, supports through DEFINITY® Release 9.
Issue 1.5	This version of the document was revised in June, 2001 to support DEFINITY® Release 9.5.
Issue 1.6	This version of the document was revised to support DEFINITY® Release 10 and the 4630 IP Telephone.
Issue 1.7	This is the current version of this document, revised to support MultiVantage™ Release 11 and the 4602 and 4620 IP Telephones.

Terms Used in This Guide

802.1p 802.1Q	802.1Q defines a layer 2 frame structure that supports VLAN identification and a QoS mechanism usually referred to as 802.1p, but the content of 802.1p is now incorporated in 802.1D.
ARP	Address Resolution Protocol, used to verify that the IP address provided by the DHCP server is not in use by another IP Telephone.
CELP	Code-excited linear-predictive; voice compression requiring only 16 kbps of bandwidth.
CLAN	Control LAN, type of TN799 circuit pack.
DHCP	Dynamic Host Configuration Protocol, an IETF protocol used to automate IP Address allocation and management.
DiffServ	Differentiated Services, an IP-based QoS mechanism.
DNS	Domain Name System, an IETF standard for ASCII strings to represent IP addresses.
IETF	Internet Engineering Task Force, the organization that produces standards for communications on the internet.
LAN	Local Area Network.
LDAP	Lightweight Directory Access Protocol, an IETF standard for database organization and query exchange.
MAC	Media Access Control, ID of an endpoint.
NAPT	Network Address Port Translation.
NAT	Network Address Translation.
PSTN	Public Switched Telephone Network, the network used for traditional telephony.
QoS	Quality of Service, used to refer to a number of mechanisms intended to improve audio quality over packet-based networks.
RRQ	Read Request packet, a message sent from the 4600 Series IP Telephone to the TFTP server, requesting to download the upgrade script and the application file.
RSVP	Resource ReSerVation Protocol, used by hosts to request resource reservations throughout a network.
RTCP	RTP Control Protocol, monitors quality of the RTP services and can provide real-time information to users of an RTP service.
RTP	Real-time Transport Protocol, provides end-to-end services for real-time data (such as voice over IP).
TCP/IP	Transmission Control Protocol/Internet Protocol, a network-layer protocol used on LANs and internets.

TFTP	Trivial File Transfer Protocol, used to provide downloading of upgrade scripts and application files to the IP Telephones.
UDP	User Datagram Protocol, a connectionless transport-layer protocol.
VLAN	Virtual LAN.

Conventions Used in This Guide

This guide uses the following textual, symbolic, and typographic conventions to help you interpret information.

Symbolic Conventions

NOTE This symbol precedes additional information about a topic. This information is not required to run your system.

CAUTION:
This symbol is used to emphasize possible harm to software, possible loss of data, or possible service interruptions.

Typographic Conventions

This guide uses the following typographic conventions:

`command`	Words printed in this type are commands that you enter into your system.
device	Words printed in this type indicate parameters associated with a command for which you must substitute the appropriate value. For example, when entering the **`mount`** command, ***device*** must be replaced with the name of the drive that contains the installation disk.
Administrative	Words printed in bold type are menu or screen titles and labels, or items on menus and screens that you select to perform a task.
italics	Italic type indicates a document that contains additional information about a topic.
<Enter>	Words enclosed in angle brackets represent a single key that should be pressed. These include **<Ctrl>**, **<Enter>**, **<Esc>**, **<Insert>**, and **<Delete>**.

Online Documentation

The online documentation for the 4600 Series IP Telephones is located at the following URL:

http://www.avaya.com/support

Related Documents

- *DEFINITY® Documentation Release 8.4*

  This CD contains documentation that describes, among other things, how to administer a DEFINITY switch with Release 8.4 software.

  This document is provided with the DEFINITY Release 8.4 product.

- *DEFINITY® Documentation Release 9*

  This CD contains documentation that describes, among other things, how to administer a DEFINITY switch with Release 9 software.

  This document is provided with the DEFINITY Release 9 product.

- *DEFINITY® Documentation Release 10*

  This CD contains documentation that describes, among other things, how to administer a DEFINITY switch with Release 10 software.

  This document is provided with the DEFINITY Release 10 product.

- *Avaya MultiVantage™ Software Documentation Release 11*

  This document describes how to administer a switch with Avaya MultiVantage™ software.

  This document is provided with the MultiVantage™ Release 11 product.

The following documents are available on the web site listed above under Online Documentation.

- *4600 Series IP Telephones Safety Instructions (for 4602/4606/4612/4620/4624/4630 IP Telephones), Issue 1, July 2002 (555-233-779)*

  This document contains important user safety instructions for the 4600 Series IP Telephones.

- *30A Switched Hub Set Up Quick Reference, Issue 2, July 2002 (Comcode 700234750; Document Number 555-236-700)*

  This document contains important safety and installation information for the 30A Switched Hub.

- *4600 Series IP Telephone Installation Guide (555-233-128)*

  This document describes how to install 4600 Series IP Telephones. It also provides troubleshooting guidelines for the 4600 Series IP Telephones.

- *4602 IP Telephone User Guide (555-233-780)*

  This document provides detailed information about using the 4602 IP Telephone.

- *4606 IP Telephone User Guide (555-233-765) (Int'l 555-233-769)*

  This document provides detailed information about using the 4606 IP Telephone.
- *4612 IP Telephone User Guide (555-233-766) (Int'l 555-233-770)*

  This document provides detailed information about using the 4612 IP Telephone.
- *4620 IP Telephone User Guide (555-233-781)*

  This document provides detailed information about using the 4620 IP Telephone.
- *4624 IP Telephone User Guide (555-233-768) (Int'l 555-233-771)*

  This document provides detailed information about using the 4624 IP Telephone.
- *4630 IP Telephone User Guide (555-233-764)*

  This document provides detailed information about using the 4630 IP Telephone.

The following documents provide standards relevant to IP Telephony.

IETF Documents

The following documents are available for free from the IETF web site: http://www.ietf.org/rfc.html.

Requirements for Internet Hosts - Communication Layers, October 1989, by R. Braden (STD 3: RFC 1122)

Requirements for Internet Hosts - Application and Support, October 1989, by R. Braden (STD 3: RFC 1123)

Internet Protocol (IP), September 1981, by Information Sciences Institute (STD 5: RFC 791), as amended by *Internet Standard Subnetting Procedure*, August 1985, by J. Mogul and J. Postel (STD 5: RFC 950)

Broadcasting Internet Datagrams, October 1984, by J. Mogul (STD 5: RFC 919)

Broadcasting Internet Datagrams in the Presence of Subnets, October 1984, by J. Mogul (STD 5: RFC 922)

User Datagram Protocol (UDP), August 28, 1980, by J. Postel (STD 6: RFC 768)

Transmission Control Protocol (TCP), September 1981, by Information Sciences Institute (STD 7: RFC 793)

Domain Names - Concepts and Facilities (DNS), November, 1987, by P. Mockapetris (STD 13: RFC 1034)

Domain Names - Implementation and Specification (DNS), November 1987, by P. Mockapetris (STD 13: RFC 1035)

The TFTP Protocol (Revision 2), (TFTP), July 1992, by K. Sollins, (STD 33: RFC 1350:) as updated by *TFTP Option Extension*, May 1998, by G. Malkin and A. Harkin (RFC 2347)

An Ethernet Address Resolution Protocol (ARP), November 1982, by David C. Plummer (STD 37: RFC 826)

Dynamic Host Configuration Protocol (DHCP), March 1997, by R. Droms (RFC 2131)

DHCP Options and BOOTP Vendor Extensions, March 1997, by S. Alexander and R. Droms (RFC 2132)

RTP: A Transport Protocol for Real-Time Applications (RTP/RTCP), January 1996, by H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson (RFC 1889)

Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, *(DIFFSRV)*, December 1998, by K. Nichols, S. Blake, F. Baker and D. Black (RFC 2474)

Introduction to version 2 of the Internet-standard Network Management Framework (SNMPv2), April 1993, by J. Case, K. McCloghrie, M. Rose, and S. Waldbusser (RFC 1441)

Management Information Base for Network Management of TCP/IP Internets: MIB-II, March 1991, edited by K. McCloghrie and M. Rose (RFC 1213)

SNMPv2 Management Information Base for the Internet Protocol using SMIv2, November 1996, edited by K. McCloghrie (RFC 2011)

Structure of Management Information Version 2 (SMIv2), April 1999, edited by K. McCloghrie, D. Perkins, and J. Schoenwaelder (RFC 2578)

Resource ReSerVation Protocol VI, September 1997, by R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin (RFC 2205)

ITU Documents

The following documents are available for a fee from the ITU web site: http://www.itu.int.

Recommendation G.711, Pulse Code Modulation (PCM) of Voice Frequencies, November 1988.

Recommendation G.729, Coding of speech at 8 kbit/s using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP), March 1996.

Annex A to Recommendation G.729: Reduced complexity 8 kbit/s CS-ACELP speech codec, November 1996.

Annex B to Recommendation G.729: A silence compression scheme for G.729 optimized for terminals conforming to Recommendation V.70, November 1996.

Recommendation H.225.0, Call signalling protocols and media stream packetization for packet-based multimedia communications systems, February 1998.

Recommendation H.245, Control protocol for multimedia communication, February 1998.

Recommendation H.323, Packet-based multimedia communications systems, February 1998.

ISO/IEC, ANSI/IEEE Documents

The following documents are available for a fee from the ISO/IEC standards web site: http://www.iec.ch.

International Standard ISO/IEC 8802-2:1998 ANSI/IEEE Std 802.2, 1998 Edition, Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks- Specific requirements- Part 2: Logical Link Control.

ISO/IEC 15802-3: 1998 ANSI/IEEE Std 802.1D, 1998 Edition, Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Common specifications- Part 3: Media Access Control (MAC) Bridges.

IEEE Std 802.1Q-1998, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.

Customer Support

For support for your 4600 Series IP Telephones, call the Avaya support number provided to you by your Avaya representative or Avaya reseller.

Information about Avaya products can be obtained at the following URL:

http://www.avaya.com/support

Overview of Voice over IP (VoIP)

2

Introduction

This chapter describes the differences between data and voice networks, and the factors that influence the performance of VoIP. The installation and administration of 4600 Series IP Telephones on DEFINITY® servers, and the installation and configuration of DHCP and TFTP are addressed.

Overview of Voice over IP

The 4600 Series IP Telephones allow enterprises to use Voice over IP (that is, packet-switched networks) instead of telephony over the Public Switched Telephone Network (PSTN). However, the use of data networks for transmitting voice packets poses the problem that data networks were not designed for the specific qualities required by voice traffic.

Data and Voice Network Similarities

Data and voice networks share similar functions due to the nature of networking.

- **Signaling** is used to establish a connection between two endpoints.

  In a voice network, signaling is used to identify who the calling party is trying to call and where the called party is on the network. Traditional telephony uses terminals with fixed addresses and establishes a fixed connection for the communication session between two such terminals, allocating fixed bandwidth resources for the duration of the call.

  IP communications constitute a connectionless network, having neither fixed addresses nor fixed connections.

- **Addressing**. Each terminal on a network must be identified by a unique address.

  In a voice network the unique address is a permanent attribute, based on international and national numbering plans, as well as those based on local telephone company practices and internal customer-specific codes.

  In IP communications, dial plans track extension numbers assigned to terminals. No fixed connection path is needed.

- **Routing** is related to addressing and allows connections to be established between endpoints.

Though these functions are common to data and voice networks, the implementations differ.

Delay and Jitter

Data traffic is generally short and comes in bursts. Data networks like the Internet were designed to manage these bursts of traffic from many sources on a first-come, first-served basis. Data packets are sent to multiple destinations, often without any attempt to keep them in a particular order.

Voice networks are designed for continuous transmission during a call. The traffic is not bursty, and the conversation uses a specific amount of bandwidth between the two ends for the duration of the call.

Several features of data networks are unsuitable for voice telephony:

- Data networks are designed to deliver data at the destination, but not necessarily within a certain time. This produces delay (latency). In data networks, delay tends to be variable. For voice messages, variable delay results in jitter, an audible chopiness in conversations.
- Variable routing also can result in loss of timing synchronization, so that packets are not received at the destination in the proper order.
- Data networks have a strong emphasis on error correction, resulting in repeated transmissions.

While data network concepts include prioritization of traffic types to give some forms of traffic greater reliability (for example, for interactive transactions), data requirements tend to be not as strict as most voice requirements.

Release 1.1 of the 4600 Series IP Telephones includes a dynamic jitter buffer. This feature automatically smooths jitter to improve audio quality.

Tandem Coding

Tandem coding (also called transcoding) refers to the conversion of a voice signal from analog to digital and back again. When calls are routed over multiple IP facilities, they may be subject to multiple transcodings. The multiple conversions between analog and digital coding result in a deterioration in the voice quality. Tandem coding should be avoided wherever possible in any compressed voice system (for example, minimizing analog trunking on the PBX).

Voice Coding Standards

There are a number of voice coding standards. The Avaya 4600 Series IP Telephones offer the options described below.

G.711, which describes the 64 kbps PCM voice coding technique. G.711-encoded voice is already in the correct format for digital voice delivery in the public phone network or through PBXs.

G.729A and G.729B, which describe adaptive code-excited linear-predictive (CELP) compression that enables voice to be coded into 8 kbps streams.

Release 1.6 of the 4600 Series IP Telephones provides support for G.711 silence suppression and custom packet loss concealment, which can improve audio quality significantly.

H.323 Standard

Internal signaling provides connection control and call progress (status) information. The H.323 standard is used for internal signaling for IP packet voice networks. H.323 defines more than simply voice. It defines a complete multimedia network (voice, video, data), with everything from devices to protocols. The H.245 standard links all the entities within H.323 by negotiating facilities among participants and H.323 network elements.

The H.323 standard makes G.711 PCM compression the default form of compression. All other compression formats are optional.

DHCP

Dynamic Host Configuration Protocol (DHCP) allows a server to assign IP addresses and other parameters to devices such as the 4600 Series IP Telephones on an as-needed basis. This eliminates the need to configure each end station with a static IP address. The DHCP application also passes information to the 4600 Series IP Telephone, identifying the IP Addresses of the PBX and the TFTP server and the path to the upgrade script and the application file on the TFTP server. For further information, refer to DHCP and TFTP Servers on page 2-7 and DHCP on page 4-6.

TFTP

During the installation and, if necessary, during the reset of the 4600 Series IP Telephones, the upgrade script and potentially, the application file, are downloaded from the Trivial File Transfer Protocol (TFTP) server to each IP Telephone, simplifying the software upgrade process. For further information, refer to DHCP and TFTP Servers on page 2-7 and TFTP on page 4-17.

NAT

A Network Address Translator is an application that can be administered between your network and the Internet. The NAT translates network layer IP addresses so your local intranet IP addresses can duplicate global, Internet addresses. A detailed discussion of NAT is beyond the scope of this document, but it should be noted that use of NAT can lead to problems affecting the consistency of addressing throughout your network. In Release 1.6 and earlier Releases of the 4600 Series IP Telephones, NAT is **not** recommended for networks handling IP-based telephony traffic. As of Release 1.7, all 4600 Series IP Telephones support NAT interworking; hence, there are no problems with NAT and Release 1.7 of the 4600 Series IP Telephones. Note, however, that support for NAT does not imply support for Network Address Port Translation (NAPT). Specifically, the 4600 Series IP Telephones do not support communication to the PBX through any NAPT device.

QoS

Quality of Service (QoS) is a term covering several initiatives to maximize the quality of the voice heard at both ends of a call that originates, terminates, or both, on an IP-based telephone. These initiatives include various prioritization schemes to offer voice packets a larger or prioritized share of network resources. These schemes include standards such as IEEE's 802.1D and 802.1Q, the Internet Engineering Task Force's (IETF's) "Differentiated Services", RTP Control Protocol (RTCP) and Resource ReSerVation Protocol (RSVP), and port-based priority schemes such as UDP port selection. Documentation for your LAN equipment will elaborate on the extent your network can support any or all of these initiatives. See Chapter 4, Server Administration for some implications of QoS for the 4600 Series IP Telephones.

As of Release 1.7, both the 4620 and 4630 IP Telephones provide information to the end user about network audio quality that may be of use to the LAN Administrator. For specific information, see QoS with 4620 and 4630 IP Telephones on page 4-22.

SNMP

Simple Network Management Protocol (SNMP) is a family of standards-based protocols and procedures to allow vendor-independent management of data networks. Using a simple set of protocol commands, an SNMP-compliant device will store information in standard format in one or more Management Information Bases (MIBs). In general, devices will support the standards-specific MIB termed MIB-II. In addition, devices may define one or more "custom MIBs" that contain information about the specifics of the device.

Release 1.1 of the 4600 Series IP Telephones is fully compatible with SNMPv2c (a later version of SNMP) and with Structure of Management Information Version 2 (SMIv2), although the telephones will respond correctly to queries from entities that comply with earlier versions of SNMP, such as SNMPv1. "Fully compatible" means that the telephones will respond to queries directed either at the MIB-II or the Custom MIB. The 4600 Series IP Telephone Custom MIB is read-only (values therein cannot be changed externally via network management tools). Similarly, although the 4600 Series IP Telephone's MIB-II has read/write permissions in accordance with the standard, to improve security any writes to MIB-II are saved but otherwise ignored.

More information about SNMP and MIBs can be found in the IETF references listed in Chapter 1, Related Documents. Appendix A of this LAN Administration Guide lists the Custom MIB, which is also available for download in *.txt format on the Avaya support website.

Network Assessment

The current technology allows optimum network configurations to deliver VoIP with perceived voice quality close to that of the Public Switched Telephone Network (PSTN). Not every network is able to take advantage of packet voice transmissions. Some data networks have insufficient residual capacity for even compressed voice traffic. In addition, the usual approach to developing data networks by integrating products from many vendors makes it necessary to test the components for compatibility with Voice over IP traffic.

It is assumed that your organization has performed a network assessment (with or without the assistance of Avaya) before attempting to install Voice over IP, in order to have a high degree of confidence that the existing data network has the capacity to carry voice packet traffic and is compatible with the required technology.

A network assessment would include a determination of the following:

- A network audit to review existing equipment and evaluate its capabilities, including its ability to meet planned voice and data needs.
- A determination of network objectives, including the dominant traffic type, choice of technologies, and setting voice quality objectives.

The assessment should leave you confident that the implemented network will have the capacity for the foreseen data and voice traffic, and can support H.323, DHCP, TFTP, and jitter buffers in H.323 applications.

It is important to distinguish between compliance with the minimal VoIP standards and support for QoS which is needed to run VoIP on your configuration.

Suggestions for Installation and Configuration

Reliability and Performance

There is a cost/performance trade-off associated with Voice over IP. Greater reliability and improved performance can be obtained through server redundancy and components with higher bandwidth capabilities.

The reliability and performance of the traditional PBX systems have been very high. Although much of the LAN is outside of the control of the PBX, there are several points to consider which enhance the reliability and performance of the IP Telephone network.

All 4600 Series IP Telephones support the tools "ping" and "traceroute." These are standard LAN/WAN tools for identifying whether two points on a network can communicate with each other, and what path a sample communication takes as it traverses the network from one point to the other. All 4600 Series IP Telephones will respond appropriately to a ping or a traceroute message sent from the DEFINITY® or MultiVantage™ switch or any other source on your network, although these telephones will not, in general, initiate a ping or traceroute. With Release 1.6 of the 4600 Series IP Telephones comes support of "remote ping" and "remote traceroute." The switch can instruct such a 4600 Series IP Telephone to initiate a ping, or a traceroute, to a specified IP address. The telephone will carry out that instruction and send a message to the switch informing it of the results. See your DEFINITY® or MultiVantage™ Administration documentation for more details.

IP Address Lists and Station Number Portability

With Release 1.5 of the 4600 Series Telephones comes the capability to specify lists of IP addresses (either dotted decimal or DNS format) for key elements of the network, rather than merely one address for each. Specifically, you can specify up to 127 total characters in each list of the following: router/gateways, TFTP servers, and the call server. When the 4600 telephone is powered up or is rebooted, it attempts to establish communication with these various network elements in turn, starting with the first address on the respective list. If the communication is denied, or times out, the telephone proceeds to the next address on the appropriate list and tries that one. The telephone does not report failure unless all the addresses on a given list have failed.

Obviously, this capability can significantly improve the reliability of IP telephony by maximizing the likelihood the telephone can communicate with backup equipment if the primary equipment is down or inaccessible (say, perhaps due to a limited network outage).

However, this capability also has the advantage of making station number portability easier. Assume a situation where the company has multiple locations (for example, London and New York), all sharing a corporate IP network. Users want to take their telephones from their offices in London and bring them to New York. When users power up their telephones in the new location, the local DHCP server will generally route them to the local switch, which denies service because it knows nothing about these new users. However, with proper administration of the local DHCP server, the telephone knows to try a second call server IP address, this one in London. The user can then be automatically registered with the London switch.

Chapter 4 contains details on administration of DCHP servers for lists of alternate call servers, router/gateways, and TFTP servers. For specific information, see DNS Addressing in Chapter 4.

Security

In VoIP, physical wire is replaced with an IP connection. The connection is more mobile. Unauthorized relocation of the IP telephone allows unauthorized users to send and receive calls as the valid owner. For further details on toll fraud, refer to the DEFINITY® or MultiVantage™ documents in Chapter 1, Related Documents.

4600 Series IP Telephones

Dual Connection Architecture

Releases 1.0 and 1.1 of the 4600 Series IP Telephones use dual connection architecture to communicate with the DEFINITY® or MultiVantage™ switch. In the dual connection architecture, two station extensions must be administered for each telephone.

Single Connection Architecture

Release 1.5 and subsequent releases of the 4600 Series IP Telephones use single connection architecture to communicate with the DEFINITY® or MultiVantage™ switch. In the single connection architecture, only one station extension must be administered for each telephone.

Registration and Authentication

The DEFINITY® or MultiVantage™ switch supports the registering and authentication of 4600 Series IP Telephones using the extension and password. For further information, see Related Documents on page 1-5.

Software

As shipped from the factory, the 4600 Series IP Telephones may not contain sufficient software for registration and operation. When the phone is first plugged in, a software download from a TFTP server is initiated. This gives the phone its proper functionality.

For downloads of software upgrades, the PBX provides the capability for a remote restart of the 4600 Series IP Telephone. As a consequence of restarting, the phone automatically restarts reboot procedures. If new software is available, a download will result.

WAN Considerations

QoS is harder on a WAN than a LAN. A LAN assumes no bandwidth concerns. A WAN assumes a finite amount of bandwidth. Therefore, QoS considerations are more significant when the IP telephony environment includes a WAN. In addition, there are administrative and hardware compatibility issues unique to WANs.

DHCP and TFTP Servers

The DHCP server provides the following information to the 4600 Series IP Telephone:

- IP address of the 4600 Series IP Telephone
- IP Address and port number of the TN799 board on the DEFINITY® or MultiVantage™ Call server. On the call server, the standard port number is 1719.

- IP Address of the TFTP server
- The subnet mask
- IP Address of the router

You should administer the LAN so that every IP Telephone can access a DHCP server with the above information.

The IP Telephone will not function without an IP address. The failure of a DHCP server at boot time will leave all the affected voice terminals unusable. (Although it is possible for the user to manually assign an IP address to an IP Telephone, when the DHCP server finally returns, the telephone will never look for a DHCP server unless the static IP data is unassigned manually. In addition, manual entry of IP data is an error-prone process.) It is therefore **strongly recommended** that a DHCP server be available when the IP Telephone reboots.

A minimum of two DHCP servers is recommended for reliability.

The TFTP server provides the 4600 Series IP Telephone with a script file and, if appropriate, new or updated application software (see Step 3, Telephone and TFTP Server under Initialization Process below). In addition, you can edit the script file to customize telephone parameters for your specific environment (see Administering Options for the 4600 Series IP Telephones in Chapter 4, Server Administration).

Initialization Process

The following is a high-level description of the information exchanged when the telephone is initializing and registering. This description, which assumes all equipment is properly administered ahead of time, may be helpful in explaining how the 4600 Series IP Telephones relate to the routers and servers in your network.

Step 1: Telephone to Network

The telephone is appropriately installed and powered, and after a short initialization process, the telephone identifies the LAN speed and sends a message out into the network, identifying itself and requesting further information. A router in the network receives this message, and relays it to the appropriate DHCP server.

Step 2: DHCP Server to Telephone

The DHCP server provides information to the telephone, as described in DHCP and TFTP Servers on page 2-7. Among other data passed to the telephone is the IP address of the TFTP server, which is crucial for the next step.

Step 3: Telephone and TFTP Server

The telephone queries the TFTP server, which transmits a script file to the telephone. This script file, at a minimum, tells the telephone which application file the telephone should be using (the application file is the software that has the telephony functionality, and can be easily updated for future enhancements).

The telephone uses the script file to determine if it has the proper application file. A newly-installed telephone will have no application file, and hence does not have the proper one. A previously-installed telephone may or may not have the proper application file. In any event, if the telephone determines it does not have the application file the script file says the telephone should have, the telephone requests a download of the proper application file from the TFTP server. The TFTP server then downloads the file and conducts some checks to ensure the file was downloaded properly. If the telephone determines it already has the proper file, it proceeds to the next step without downloading the application file again.

Step 4: Telephone and the DEFINITY®/MultiVantage™ Call Server

In this step, the telephone and the PBX exchange a series of messages which cause the display on the telephone to prompt the user. For a new installation, the user must enter the telephone's extension and the call server password. For a restart of an existing installation, this information is already stored on the telephone, but the user must confirm the information. In either case, manual intervention is required. The telephone and the switch exchange more messaging, with the expected result that the telephone is appropriately registered on the switch.

More details about the installation process are available in the *4600 Series IP Telephone Installation Manual* and in Chapter 3 of this document.

Requirements

3

Introduction

The 4600 Series IP Telephones use Internet Protocol (IP) technology with Ethernet line interfaces. The IP telephones supplement the existing DEFINITY®/MultiVantage™ IP Solutions platform. This feature provides the user with the capability to natively administer and maintain the new 4600 Series IP Telephones.

The 4600 Series IP Telephones provide support for DHCP and TFTP over IPv4/UDP which enhance the administration and servicing of the phones. These phones use DHCP to obtain dynamic IP addresses and TFTP to download new versions of software for the phones.

The 4600 Series IP Telephones provide the ability to have one connection on the desktop for both the telephone set and the PC using the telephone's built-in hub.

Hardware Requirements

Before plugging in the 4600 Series IP Telephone, verify that all of the following requirements have been met. Failure to do so will prevent the telephone from working and may have a negative impact on your network.

The following hardware is required for 4600 Series IP Telephones to work properly.

- The DEFINITY® switch must be installed and administered correctly, with Release 8.4 or later.
  - — DEFINITY® Release 8.4 supports the 4612 and 4624 IP Telephones.
  - — DEFINITY® Release 9 and later support the 4606, 4612, and 4624 IP Telephones.
  - — DEFINITY® Release 10 and later support the 4630 IP Telephone.
  - — MultiVantage™ Release 11 supports the 4602 and 4620 IP Telephones.

NOTE For Release 1.1 of any of the 4600 Series IP Telephones, the DEFINITY® switch must have Release 9 software installed. For Release 1.5 of any of the 4600 Series Telephones, the DEFINITY® switch must have Release 9.5 software. For Release 1.6 of the 4600 Series IP Telephones, and for support of the 4630 IP Telephone, the DEFINITY® switch must have Release 10 software. For Release 1.7 of the 4600 Series IP Telephones, and for support of the 4602 and 4620 IP Telephones, the MultiVantage™ switch must have Release 11 software.

- The following two circuit packs must be resident on the PBX server used to support the IP telephones:
  - TN2302 AP Media Processor circuit pack converts the audio levels for the IP telephone to audio levels for DCP phones when IP phones are used in a call with non-IP telephones.
  - TN799 Control-LAN (CLAN) circuit pack for the signaling capability (either the B or C vintage) on the csi, si, and r platforms.
- A Category 5 LAN. If the telephones are to be powered from the LAN, the LAN must comply with the IEEE 802.3af standard for LAN powering.
- Electrical power provided to each phone by one of the following two sources.
  - A Telephone Power Module (DC power jack) (must be ordered separately).
  - IEEE 802.3af, if the LAN supports this powering scheme (although the 4630 **cannot** be powered this way).
- Verify that the 4600 Series IP Telephone package includes the following components:
  - 1 telephone set.
  - 1 AB1C handset.
  - 1 H4DU 9-foot long (when extended) 4-conductor coiled handset cord, plugged into the telephone and the handset.
  - 1 Category 5 modular line cord for the connection between the IP Telephone and the Ethernet wall plug.
  - Non-system-specific safety and installation documentation.
  - Power brick
  - Stylus (4630 IP Telephones only)
- You may need a Category 5 modular line cord for the connection from the 4600 Series IP Telephone to the PC.

Refer to the *4600 Series IP Telephone Installation Guide*.

The IP telephones work the same on all DEFINITY®/MultiVantage™ platforms.

Software Requirements

The following software is required for 4600 Series IP Telephones to work properly.

- DEFINITY®/MultiVantage™ Release 8.4 or later software, as appropriate. The H.323 and IP station customer options must be turned on.
- The DHCP server and application should be installed and properly administered, as described in DHCP on page 4-6.

  ⚠ WARNING:

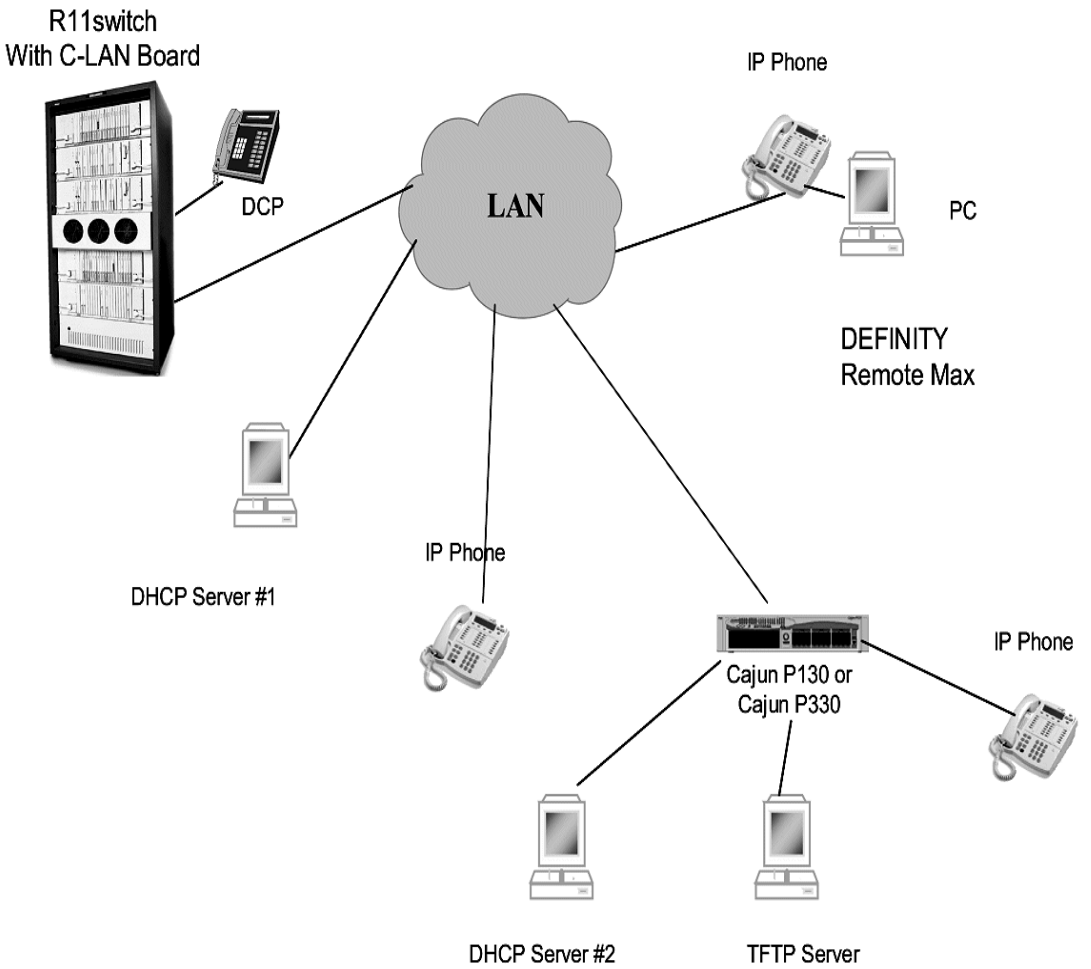
  A DHCP server is not mandatory, but static addressing is necessary when a DHCP server is unavailable.
  Due to the difficulties associated with static addressing, it is very strongly recommended that a DHCP server be installed and that static addressing be avoided.

- The TFTP server and application must be installed and properly administered, as described in TFTP on page 4-17.
- For 4630 IP Telephone environments, if users are to have access to LDAP directories or corporate Websites, the appropriate servers must be in place, and the 4630 telephones must be appropriately administered in accordance with Chapter 4, Server Administration.

Ensure that all required parameters are configured correctly. For DEFINITY/MultiVantage, see your administration documentation. For the DHCP and TFTP servers, see Chapter 4, Server Administration.

Figure 3-1 on the following page illustrates a sample configuration using the 4600 Series IP Telephones.

Figure 3-1. Sample Configuration Using 4600 Series IP Telephones



Server Administration

4

Introduction

When a 4600 Series IP Telephone is plugged in and powered, it automatically negotiates with its associated LAN to determine the Ethernet speed. From that point on, the actions taken by the phone depend in large part on the action taken in the network before the phone is installed, and on the actions taken, if any, by the installer. This chapter discusses in detail the parameters and other data the telephone needs to operate and the alternatives for delivering that information to the telephone, where appropriate. Recommendations and specifications on which alternative(s) to choose for which parameter(s) are also provided.

The parameters under which the phone needs to operate are summarized as follows:

- Telephone Administration on the Call Server
- IP address management for the telephone
- Tagging Control and VLAN administration for the telephone, if appropriate
- Quality of Service (QOS) administration for the telephone, if appropriate
- Site-specific Option Number (SSON) setting of DHCP servers, if appropriate
- Interface administration for the telephone, if appropriate
- Application-specific administration for the telephone, if appropriate (for example, Directory- or Web-specific information required for these optional 4630 applications)

The delivery mechanisms are:

- Maintaining the information on the Call Server (for example, the DEFINITY®/MultiVantage™ switch)
- Manually entering the information via the telephone dialpad
- Administering the DHCP Server
- Editing the script files on the TFTP Server

These parameters can be administered in a variety of ways, as indicated in Table 4-1, below. Note that not all parameters can be administered on all delivery mechanisms.

Table 4-1. Administration Alternatives and Options for 4600 Series IP Telephones

Parameter(s)	Administrative Mechanisms	For More Information See:
Telephone Administration	Call Server	Administering 4600 Series IP Telephones on DEFINITY®/ MultiVantage™ Servers on page 4-4 and Related Documents on page 1-5.
IP Addresses	DHCP (strongly recommended)	DHCP and TFTP on page 4-4, especially DHCP on page 4-6.
	TFTP Script files	DHCP and TFTP on page 4-4 and Administering Options for the 4600 Series IP Telephones on page 4-24.
	Manual administration at the phone	See "Static Addressing Installation" in Chapter 3 of the *4600 IP Telephone Installation Guide.*
Tagging and VLAN	DHCP	DHCP and TFTP on page 4-4, and Administering Options for the 4600 Series IP Telephones on page 4-24.
	TFTP Script files (strongly recommended)	DHCP and TFTP on page 4-4 and Administering Options for the 4600 Series IP Telephones on page 4-24.
	Manual administration at the phone	See "Static Addressing Installation" in Chapter 3 of the *4600 IP Telephone Installation Guide*.
Quality of Service	DHCP	DHCP and TFTP on page 4-4, and Administering Options for the 4600 Series IP Telephones on page 4-24.
	TFTP Script files (strongly recommended)	DHCP and TFTP on page 4-4, and Administering Options for the 4600 Series IP Telephones on page 4-24.
	Manual administration at the phone	See "QOS Option Setting" in Chapter 3 of the *4600 IP Telephone Installation Guide*.
Interface	DHCP	DHCP and TFTP on page 4-4, and Administering Options for the 4600 Series IP Telephones on page 4-24.

Parameter(s)	Administrative Mechanisms	For More Information See:
	TFTP Script files (strongly recommended)	DHCP and TFTP on page 4-4, and Administering Options for the 4600 Series IP Telephones on page 4-24.
	Manual administration at the phone	See "Secondary Ethernet (Hub) Interface Enable/Disable" in Chapter 3 of the *4600 IP Telephone Installation Guide*.
SSON	DHCP	Customizing the Site-Specific Option Number (SSON) on page 4-25; DHCP and TFTP on page 4-4, especially DHCP on page 4-6.
	TFTP Script files (strongly recommended)	Customizing the Site-Specific Option Number (SSON) on page 4-25; DHCP and TFTP on page 4-4, especially TFTP on page 4-17.
	Manual administration at the phone	See "Site-Specific Option Number Setting" in Chapter 3 of the *4600 IP Telephone Installation Guide*.
Application-specific parameters	DHCP	DHCP and TFTP on page 4-4, especially DHCP on page 4-6. Also, Customizing the 4630 IP Telephone on page 4-26 and Customizing the 4620 IP Telephone on page 4-32.
	TFTP Script files (strongly recommended)	DHCP and TFTP on page 4-4, especially TFTP on page 4-17. Also, Customizing the 4630 IP Telephone on page 4-26 and Customizing the 4620 IP Telephone on page 4-32.

General information about administering DHCP servers is covered in DHCP and TFTP on page 4-4, especially DHCP on page 4-6. General information about administering TFTP servers is covered in DHCP and TFTP on page 4-4, especially TFTP on page 4-17. Once you are familiar with that material, you will be able to administer options on the telephone in accordance with Administering Options for the 4600 Series IP Telephones on page 4-24.

Administering 4600 Series IP Telephones on DEFINITY®/MultiVantage™ Servers

DEFINITY® Release 8.4

DEFINITY® Release 8.4 supports the 4612 and 4624 IP Telephones. The 4612 and 4624 IP Telephones are aliased as 6424 telephones, administered as IP Softphones. The administrative forms for the 6424 IP Softphone are used for the two IP Telephones. See Related Documents on page 1-5. Follow these guidelines:

- Alias the IP Telephone as a 6424D+ DCP set, with the **IP Softphone** field set to "y."
- Administer a **Media Complex Ext** for the audio channel.

DEFINITY® Releases 9, 9.5, 10, and MultiVantage™ Release 11

DEFINITY® Releases 9 and 9.5 provide support for the 4606, 4612, and 4624 IP Telephones. DEFINITY® Release 10 adds support for the 4630 IP Telephone. MultiVantage™ Release 11 adds support for the 4602 and 4620 IP Telephones. Administration of a 4612 and 4624 IP telephone is identical to a 6424 IP softphone. See Related Documents on page 1-5. Follow these guidelines:

- On the Customer Options form, verify that the **IP Stations** field is set to "y." If it is not, contact your Avaya sales representative.
- The **IP Softphone** field does not have to be set to "y."

DHCP and TFTP

Dynamic Host Configuration Protocol (DHCP) provides a means by which configuration parameters can be automatically assigned to clients on a TCP/IP network. This minimizes the maintenance of a network of 4600 Series IP telephones by removing the need to assign and maintain IP addresses and other parameters for each IP telephone on the network individually.

Software Checklist

Please make sure that you have purchased and/or own licenses to install and use the DHCP server and TFTP server software.

NOTE It is possible to install both the DHCP server and the TFTP server on the same machine.

WARNING:

*The circuitry in the 4600 Series IP Telephones reserves IP addresses of the form **192.168.2.x** for internal communications. The telephone(s) will not properly use addresses you specify if they are of that form.*

Required Network Information

DHCP is the control point where an enterprise controls its IP Telephones. Before administering DHCP and TFTP, complete the information in Table 4-2 below to ensure that you have the necessary information regarding your network. There may be more than one Gateway, TFTP server, subnet mask and CLAN in your configuration. You will need a copy of this table for each DHCP server.

Release 1.5 of the 4600 Series Telephones supports the ability to specify a list of IP addresses for a gateway/router, TFTP server, and DEFINITY® CLAN board(s), as explained in Chapter 2, IP Address Lists and Station Number Portability Each list may contain up to 127 total ASCII characters, with IP addresses separated by commas with no intervening spaces.

When specifying IP addresses for the TFTP server or call server, you can use either dotted decimal format ("xxx.xxx.xxx.xxx") or DNS names to identify the address(es). If you use DNS, note that the system value DOMAIN will be appended to the IP addresses you specify. If DOMAIN is null, the DNS names must be fully qualified, in accordance with IETF RFCs 1034 and 1035. For more specific information about DNS, see DHCP Generic Setup (page 4-7) and DNS Addressing (page 4-24).

Table 4-2. Required Network Information Before Installation - Per DHCP Server

1.	Gateway (router) IP address(es)
2.	TFTP server IP address(es)
3.	Subnet mask
4.	DEFINITY CLAN IP address(es)

Table 4-2. Required Network Information Before Installation - Per DHCP Server

5. DEFINITY CLAN port	Although this may be a value between 0 and 65535, the default value is 1719 and **should not** be changed unless this conflicts with an existing port assignment.
6. TFTP server file path	
7. Telephone IP address range *From:* *To:*	
8. DNS Server address(es)	If applicable

The **TFTP server file path** is the "root" directory used for all transfers by the server. This is the default directory which all files will be uploaded to or downloaded from. In configurations where the upgrade script and application files are in the default directory, item 6 should not be used.

DHCP

This section provides information on possible DHCP servers and generic information on administering a DHCP server.

Choosing a DHCP Configuration

A discussion on how to best set up your network to work with the 4600 Series IP Telephones is beyond the scope of this document. See Network Assessment on page 2-5. This document concentrates on the simplest case of the single LAN segment. Information provided here can be extrapolated for more complex LAN configurations.

⚠ WARNING:

Before you start, it is important that you understand your current network configuration. An improper installation can cause network failures or reduce the reliability and performance of your network.

DHCP Software Alternatives

Two DHCP software alternatives are common to Windows operating systems:

- Windows NT 4.0 DHCP Server
- Windows 2000 DHCP Server

Any other DHCP application may work.

It is the customer's responsibility to install and configure the DHCP server correctly. This document is limited to describing a generic administration that will work with the 4600 Series IP Telephones.

DHCP Generic Setup

Set up of a DHCP server involves the following phases:

1. Install the DHCP server software according to vendor instructions
2. Configure the DHCP server with the following information
   - IP addresses available for the 4600 Series IP Telephones
   - Lease duration (Infinite is recommended.)
   - The following DHCP options:
     - Gateway (router) IP address(es) (Table 4-2, item 1). If more than one address is listed, the total list may contain up to 127 total ASCII characters, with IP addresses separated by commas with no intervening spaces.
     - Subnet mask (Table 4-2, item 3).
     - Option 6 (DNS server(s) address list). If more than one address is listed, the total list may contain up to 127 total ASCII characters, with IP addresses separated by commas with no intervening spaces. At least one address in Option 6 must be a valid, non-zero, dotted decimal address - otherwise, DNS will fail.
     - Option 15 (DNS Domain Name). This string should contain the domain name to be used when DNS names in system parameters are resolved into IP addresses. This domain name is appended to the DNS name before the 4600 IP Telephone attempts to resolve the DNS address. Option 15 is necessary if you wish to use a DNS name for the TFTP server; otherwise, you may specify a DOMAIN as part of TFTP customization, as indicated in DNS Addressing on page 4-24.
     - Option 66 (TFTP Server Name). Note: Microsoft DHCP servers support only dotted-decimal format for TFTP addresses, not symbolic names. Option 66 need not be used if the TFTP server is identified in the Site Specific Option string (Option 176). However, to simplify configuration, we recommend that you use Option 66. If you use both Option 66 and Option 176 to identify TFTP servers, the value(s) in Option 176 will override the value(s) in Option 66.
     - A 4600 Series IP Telephone-specific DHCP option specifying information, such as TFTP server and DEFINITY® CLAN IP addresses. Use the site-specific option (SSON) at #176. The value for this option should be set to either of the following strings:

       MCIPADD=xxx.xxx.xxx.xxx,MCPORT=yyyy,TFTPSRVR=zzz.zzz.zzz.zzz,TFTPDIR=<path>

       OR

MCIPADD={list of DNS names},MCPORT=yyyy,TFTPSRVR={list of DNS names},TFTPDIR=<path>

NOTE In configurations where the upgrade script and application files are in the default directory, the TFTPDIR=<path> should not be used.

NOTE You do not have to use Option 176. For example, if the DNS server is specified in Option 6, and the Domain Name is specified in Option 15, you can use the configured names "AvayaTFTPServer" and "AvayaCallServer" for TFTPSRVR and MCIPADD, respectively.

NOTE The Call Server Name, TFTP Server Name, and SMTP Server Name must each be no more than 32 characters in length.

Where xxx.xxx.xxx.xxx is one or more IP addresses for DEFINITY®/MultiVantage™ CLAN IP boards, yyyy is the DEFINITY/MultiVantage CLAN port (1719), zzz.zzz.zzz.zzz is one or more IP addresses for TFTP servers, and <path> is the location of the location of the upgrade script and application files on the TFTP server as entered in Table 4-2, items 4, 5, 2, and 7, respectively. Each list may contain up to 127 total ASCII characters, with IP addresses separated by commas with no intervening spaces, and with quotes on either end (see the example in the **NOTES** below). If you use DNS, note that the system value DOMAIN will be appended to the IP addresses you specify. If DOMAIN is null, the DNS names must be fully qualified. See Administering Options for the 4600 Series IP Telephones on page 4-24.

NOTES

- Examples of good DNS administration include the following:
  - Option 6: "aaa.aaa.aaa.aaa"
  - Option 15: "dnsexample.yourco.com"
  - Option 66: "tftpserver.yourco.com,zzz.zzz.zzz.zzz"
  - Option 176: "MCIPADD=xxxx.xxx.xxx.xxx"
- Depending on the DHCP application you choose, you should be aware of the fact that the application most likely will not immediately recycle expired DHCP leases. An expired lease may remain reserved for the original client for a day or more (for example, Windows NT DHCP reserves expired leases for about one day). The intent of this reservation period is to protect a client's lease in case the client and the DHCP server are in two different time zones, the computers' clocks are not in synch, or the client is not on the network when the lease expires.
- The implication of this fact may be seen in the following example: Assume 2 IP addresses (hence 2 possible DHCP leases) and 3 IP telephones, two of which are using the 2 available IP addresses. When the lease expires for the first two telephones, the third will not be able to get a lease (even if the other two telephones have been removed from the network), until the reservation period expires.

The 4600 Series IP Telephone sets the indicated system values to the values of the indicated fields of the DHCPACK message as indicated in Table 4-3.

Table 4-3. DHCPACK Setting of System Values

System Value	Set to
IPADD	The yiaddr field.
NETMASK	Option #1 (if received).
GIPADD	The first four octets of option #3 (if received).
TFTPSRVR	The first four octets of the siaddr field.

The remainder of this section describes some common DHCP servers.

Windows NT 4.0 DHCP Server

This section contains details on how to verify and configure the DHCP server included in the Windows NT 4.0 server operating system.

Use Verifying the Installation of the DHCP Server below to verify whether the DHCP server is installed. If it is not, install the DHCP server. If it is installed, go to the section Initial Configuration on page 4-9 and the following section.

Verifying the Installation of the DHCP Server

Use the following procedure to verify whether the DHCP server is installed.

1. Select **Start->Settings->Control Panel**.
2. Double-click the **Network** icon.
3. Verify that **Microsoft DHCP Server** is listed as one of the Network Services on the Services Tab.
4. If it is, go to the section Initial Configuration below. If it is not, then install the DHCP server.

Initial Configuration

The Windows NT 4.0 DHCP server configuration involves setting up a scope for the IP telephone. A DHCP scope is essentially a grouping of IP devices (in this case IP telephones) running the DHCP client service in a subnet. The scope is used to define parameters for each subnet. Each scope has the following properties:

- A unique subnet mask used to determine the subnet related to a given IP address.
- A scope name assigned by the administrator when the scope is created.
- Lease duration values to be assigned to DHCP clients with dynamic addresses.

In addition, the DHCP server can assign configuration parameters to a client, and these can be specified for each individual DHCP scope.

Setting up of the Windows NT 4.0 DHCP server, requires the following steps.

1. Creating a DHCP scope for the IP Telephones.
2. Editing custom options.
3. Adding the DHCP options.
4. Activating the new scope.

Creating a DHCP Scope for the IP Telephones

Use the following procedure to create a DHCP scope for the IP Telephones.

1. Select **Start->Programs->Admin Tools->DHCP Manager**.
2. Expand **Local Machine** in the DHCP Servers window by double clicking on it until the **+** sign changes to a **-** sign.
3. Select **Scope->Create**.
4. Define the range of IP addresses used by the IP telephones listed in Line 7 of Table 4-2.

   The Start Address should be the first IP address to be used for the IP telephones.

   The End Address should be the last IP address to be used for the IP telephones.

   Subnet Mask should be set to the value as recorded in Table 4-2.

   Perform the following steps to exclude any IP addresses that you do not want to be assigned to IP telephones within the range specified by the Start and End Addresses.

   a. Enter the first IP address in the range that you would like to exclude in the Start Address field under Exclusion Range.

   b. Enter the last IP address in the range that you would like to exclude in the End Address field under Exclusion Range.

   c. Click the **Add** button.

   d. Repeat steps a. through c. for each IP Address range that you would like to exclude.

   Example

   Suppose the range of IP addresses available for your IP telephone network are:

   - 135.254.76.7 to 135.254.76.80
   - 135.254.76.90 to 135.254.76.200
   - 135.254.76.225 to 135.254.76.230

   Your start address and end address should then be 135.254.76.7 and 135.254.76.230 respectively.

You should exclude the ranges 135.254.76.81 to 135.254.76.89 and 135.254.76.201 to 135.254.76.224.

NOTE We recommend that you provision the 4600 Series IP Telephones with sequential IP addresses.

5. Under **Lease Duration**, select the **Limited To** option and set the lease duration to the maximum.
6. Enter a sensible name for the Name field, such as “DEFINITY IP Telephones.”
7. Click **OK**.

   A dialog box prompts you: **Activate the new scope now**?
8. Click **No**.

NOTE You will activate the scope when all options have been set.

Editing Custom Options

Use the following procedure to edit custom options:

1. Select **DHCP Options->Defaults** in the menu.
2. Click the **New** button.
3. Enter “46XXOPTION” for your custom in the Add Option Type dialog.
4. Select **Data Type of String** and enter **176** in the Identifier field.
5. Click the **OK** button.

   The DHCP Options menu is displayed.
6. Select the **Option Name** for 176 and set the **value string**.
7. Click the **OK** button.
8. Select **003 Router** from the drop-down list for the Option Name field.
9. Click **Edit Array**.
10. Enter the Gateway IP address recorded in Line 1 of Table 4-2 for the New IP Address field.
11. Select **Add** and then **OK**.

Adding the DHCP Option

Use the following procedure to add the DHCP option:

1. Highlight the scope you just created.
2. Select **Scope** under DHCP OPTIONS.
3. Select the 176 option that you created from Unused Option List.

4. Click the **Add** button.
5. Select option 003 from the Unused Options List.
6. Click the **Add** button.
7. Click the **OK** button.
8. Chose the Global parameter under DHCP Comments.
9. Select the 176 option that you created from Unused Option List.
10. Click the **Add** button.
11. Click the **OK** button.

Activating the Leases

Use the following procedure to activate the leases:

1. Click **Activate** under the Scope Menu.

   The light-bulb icon for the scope is lit up.

Verifying Your Configuration

This section describes how to verify that the 46XXOPTIONs are correctly configured for the Windows NT 4.0 DHCP server.

Verify the Default Option, 176 46XXOPTION

Use the following procedure to verify the default option:

1. Select **Start->Programs->Admin Tools->DHCP Manager**.
2. Expand "Local Machine" in the DHCP Servers window by double clicking on it until the + sign changes to a - sign.
3. In the DHCP Servers frame, click the scope for the IP Telephone.
4. Select **Defaults** from the **DHCP_Options** menu.
5. In the Option Name pull-down list, select **176 46XXOPTION**.
6. Verify that the Value String box contains the correct string from DHCP Software Alternatives on page 4-6.

   If not, update the string and click the **OK** button twice.

Verify the Scope Option, 176 46XXOPTION

Use the following procedure to verify the scope option:

1. Select **Scope** under DHCP OPTIONS.
2. In the *Active Options:* scroll list, click on **176 46XXOPTION**.
3. Click the **Value** button.

4. Verify that the Value String box contains the correct string from DHCP Generic Setup on page 4-7.

   If not, update the string and click the **OK** button.

Verify the Global Option, 176 46XXOPTION

1. Select **Global** under DHCP OPTIONS.
2. In the **Active Options:** scroll list, click **176 46XXOPTION**.
3. Click the **Value** button.
4. Verify that the Value String box contains the correct value from DHCP Generic Setup on page 4-7.

   If not, update the string and click the **OK** button.

Windows 2000 DHCP Server

This section describes the configuration of the DHCP server in Windows 2000.

Verifying the Installation of the DHCP Server

Use the following procedure to verify whether the DHCP server is installed:

1. Select **Start->Program->Administrative Tools->Computer Management**.
2. Under Services and Applications in the Computer Management tree, you should find **DHCP**.
3. If DHCP is not installed, install the DHCP server; otherwise please skip directly to Creating and Configuring a DHCP Scope for instructions on server configuration.

Creating and Configuring a DHCP Scope

Use the following procedure to create and configure a DHCP scope:

1. Select **Start->Programs->Administrative Tools->DHCP**.
2. In the console tree, click the DHCP server to which you wish to add the DHCP scope for the IP telephones. Typically this will simply be the name of your DHCP server machine.
3. Select **Action->New Scope** from the menu.

   Windows displays the New Scope Wizard to guide you through rest of the setup.
4. Click the **Next** button.

   The Scope Name dialog box is displayed.
5. Enter a name for the scope in the Name field, such as “DEFINITY IP Telephones,” and a brief comment in the Description field.
6. Click the **Next** button when finished.

   The IP Address Range dialog box is displayed.

7. Define the range of IP addresses used by the IP telephones listed in item 7 in Table 4-2· The Start IP Address should be the first IP address available to the IP telephones. The End IP Address should be the last IP address available to the IP telephones.
8. You may define the subnet mask in one of two ways:
   - The number of bits of an IP address to use for the network/subnet IDs.
   - The subnet mask IP address.

   Enter only one of these values. Click the **Next** button when finished.

   The Add Exclusions dialog box is displayed.
9. Exclude any IP addresses in the range specified in the previous step that you do not wish to be assigned to an IP telephone.
   a. Enter the first IP address in the range that you would like to exclude in the Start Address field under Exclusion Range.
   b. Enter the last IP address in the range that you would like to exclude in the End Address field under Exclusion Range.
   c. Click the **Add** button.
   d. Repeat steps a. through c. for each IP Address range that you would like to exclude.

   NOTE You may add additional exclusion ranges later by right clicking on the Address Pool under the newly created scope and select the New Exclusion Range option.

   Example:

   Suppose the ranges of IP addresses available for your IP telephone network are:
   - 135.254.76.7 to 135.254.76.80
   - 135.254.76.90 to 135.254.76.200
   - 135.254.76.225 to 135.254.76.230

   Your Start IP Address and End IP Address entered on the IP Address Range dialog box should then be 135.254.76.7 and 135.254.76.230 respectively.

   On the Add Exclusions dialog box, you should exclude the following ranges:
   - 135.254.76.81 to 135.254.76.89
   - 135.254.76.201 to 135.254.76.224

   Click the **Next** button when all the exclusions have been entered.

   The Lease Duration dialog box is displayed.
10. Enter 30 days in the lease duration for all telephones that will receive their IP addresses from the server. This is the duration after which a device's IP address expires and needs to be renewed by the device.
11. Click the **Next** button.

    The Configure DHCP Options dialog box is displayed.

12. Click the **No, I will activate this scope later** radio button.

    The Router (Default Gateway) dialog box is displayed.

13. For each router or default gateway, enter the IP address and click the **Add** button.

    When you are done, click the **Next** button.

    The Completing the New Scope Wizard dialog box is displayed.

14. Click the **Finish** button.

    You new scope is added under your server in the DHCP tree. It is not yet active and will not assign IP Addresses.

15. Highlight the newly created scope and select A**ction->Properties** from the menu.

16. Under Lease duration for DHCP clients, select **Unlimited** and then click the **OK** button.

⚠ WARNING:

IP Address leases are kept active for varying periods of time. To avoid having calls terminated suddenly, make the lease duration unlimited.

Adding DHCP Options

Use the following procedure to add DHCP options to the scope you created in the previous procedure:

1. On the DHCP window, right-click the "Scope Options" folder under the scope you created in the last procedure.

   A drop-down menu is displayed.

2. Click the **Configure Options...** option.

   The Scope Options dialog box is displayed.

3. In the General tab page, under the Available Options, check the 066 'Boot Server Host Name' Options checkbox.

   The String Value dialog box is displayed.

4. Enter the TFTP Server address(es) in the string value. Use the same TFTPSRVR *value* format as discussed in the TFTP Generic Setup section. For example, if you had a TFTP server at IP address *zzz.zzz.zzz.zzz* and a second TFTP server at address tftpserver.yourco.com, in the string value enter:

   "*zzz.zzz.zzz.zzz*,tftpserver.yourco.com"

5. Also under the Available Options, check the **176 Site-Specific Options** checkbox.

6. Click the **Add** button and then the **Edit Array** button.

   The IP Address Array Editor dialog box is displayed.

7. Enter the IP Address(es) for the TFTP Server(s) supporting the IP Telephones.

8. Click the **OK** button.

   The Predefined Options and Values dialog box is displayed.

9. Click the **OK** button.

   The Predefined Options and Values dialog box is closed, leaving the DHCP dialog box enabled.

10. Expand the newly created scope to reveal its Scope Options.
11. Click **Scope Options** and select **Action->Configure Options** from the menu.
12. In the General tab page, under the Available Options, check the **176 Site-Specific Options** checkbox.
13. In the Data Entry box, enter the DHCP IP telephone option string as described in DHCP Generic Setup on page 4-7.

    NOTE You can enter the text string directly on the right side of the Data Entry box under the ASCII label.

14. From the list in Available Options, check option **003 Router**.
15. Enter the gateway (router) IP address as recorded in the IP Address field of Table 4-2.
16. Click the **Add** button.
17. Click the **OK** button.

Activating the New Scope

Use the following procedure to activate the new scope.

1. In the DHCP console tree, click the IP Telephone Scope created.
2. From the Action menu, select **Activate**.

   The small red down arrow over the scope icon disappears, indicating that the scope has been activated.

TFTP

This section describes how to set up a TFTP server for downloading software updates to the 4600 Series IP Telephones.

WARNING:

The files defined by the TFTP server configuration have to be accessible from all IP Telephones. Ensure that the filenames match the names in the upgrade script, including case, since UNIX systems are case sensitive.

NOTE You can use any TFTP application you want. However, we strongly recommend using the TFTP application available for free download at www.avaya.com/support. The site also contains instructions for installing and configuring the Avaya TFTP server.

TFTP Generic Setup

The following phases are involved in setting up a TFTP server.

- Install the TFTP server software. The section below describes how to install and configure Avaya's TFTP application.
- Configure the file path parameter to the directory where the files are to be stored. This is the file path in line 6 of Table 4-2. For increased security, it is also recommended that you disable the ability to upload to the server. Note that this option may be not available to all TFTP servers.
- In addition, you may wish to enable the transfer size option (tsize) if your TFTP server supports it. This will allow the IP telephone to display the progress of the transfer by displaying the total number of data blocks.
- Download the upgrade script file and application file from the Avaya website (www.avaya.com/support) to the directory as specified by the file path.

Avaya TFTP (Suite Pro)

Configuration

Use the following procedure to configure the Avaya TFTP server:

1. Run the TFTP Suite Pro server by selecting **Start->Programs->Avaya TFTP Server >TFTPServer32**.

   The TFTP server is started.

   WARNING:

   You must re-start Avaya TFTP manually every time you reboot your TFTP server machine.

2. Select **System->Setup**.
3. Enter the following values:

   On the Outbound tab page: 1.

   The Outbound path should be the TFTP file path as recorded in Table 4-2, Required Network Information Before Installation - Per DHCP Server.

   The Enable Path options should be checked.

   Under the Options tab page: turn on the No Incoming option.

   Under the Client Limits tab page: Set the Maximum Simultaneous Clients to infinite by dragging the slide bar all the way to the right.
4. Place the *46xxupgrade.scr* file in the file path directory. (The filename *46xxupgrade.scr* is an example, not the filename you will use. See Contents of the Upgrade Script on page 4-19.)

4600 Series IP Telephone Scripts and Application Files

The files necessary to operate the 4600 Series IP Telephones are available on the following Avaya web site:

www.avaya.com/support

Three files are needed:

- A boot file, contained in the telephone when it is shipped from the factory. This file contains the software that allows the telephone to connect to the LAN and attempt to acquire necessary files and connectivity.
- An upgrade script file, which at a minimum tells the telephone which application file it should be using. The upgrade script file can also be used to point to a file you create to specific non-default settings for key telephone parameters.
- An application file, which contains all the telephony functionality of the telephone.

All three files are available from the Avaya website. In general, you do not need to download the boot file, unless Avaya produces an enhancement to the file. The upgrade script and application files always must be downloaded to the TFTP server.

The Avaya-provided default script file is sufficient to allow the telephone to use default settings, or settings changed via DHCP or local programming on the telephone dialpad. However, to use TFTP to customize values of the telephone parameters, you must create an ASCII file that resides in the same directory as the upgrade script file mentioned above. This file must be called **46xxsettings.scr** or **46xxxsettings.txt** (see **NOTE** below). This file must consist of SET commands, one per line, in this format:

SET parameter_name value

where **parameter_name** and **value** are as defined in Tables 4-4, 4-5, and 4-6. Invalid values cause the specified **value** to be ignored for the associated **parameter_name**; the default value is retained..

NOTE Most Windows systems interpret the file extension *.scr as a screen saver. The 4600 IP Telephones originally used *.scr to indicate a script file, but starting with Release 1.7, the file name can also have the extension *.txt.

NOTE If you choose to create a custom parameter file, you can edit the last line of the Avaya-provided default script file to be GET **46xxxsettings.txt**.

NOTE All values should be text strings, even if the data itself is numeric, a dotted-decimal IP address, etc.

NOTE You are encouraged not to alter the Avaya-provided upgrade script file. If Avaya changes the file in the future, to accommodate new features or functionality, any changes you might have made would be lost. You are strongly encouraged to use the **46xxsettings.scr** file to manage your customization.

NOTE The GET command causes the telephone to use TFTP to attempt to download the file specified in the GET command. This file is assumed to be at the same directory as the current upgrade script file. If the file specified in the GET command is successfully obtained, its contents are interpreted as an additional script file - that is how your options are set. If the file cannot be obtained, interpretation of the rest of the current script file is terminated (which is why we indicate the GET should be at the end of the script file). You can change the name of the file in the GET command; just ensure the target file is identically named and is located appropriately.

Table 4-4 lists the parameters, descriptions, and acceptable values for all 4600 Series IP Telephones. Table 4-5 lists the parameters, descriptions, and acceptable values specific to applications on the 4630 IP Telephone. Table 4-6 lists the parameters, descriptions and acceptable values specific to applications on the 4620 IP Telephone.

Contents of the Upgrade Script

The following is a sample upgrade script file:

NOTE The filenames 4624_000301.BIN, 4606_000301.BIN, and 4630_00301.BIN used in this sample are examples only. The names do not match those that are used in production.

```
IF $BOOTNAME SEQ 46XXCOMMON.V03 GOTO GETAPP
SET APPNAME REPLBOOT_V3.APP
GOTO END
```

```
#GETAPP
IF $MODEL4 SEQ 4630D01A GOTO SCREENSETS
IF $MODEL4 SEQ 4624D01A GOTO BIGSETS
IF $MODEL4 SEQ 4612D01A GOTO BIGSETS
IF $MODEL4 SEQ 4606D01A GOTO SMALLSETS


#SCREENSETS
SET APPNAME 4630_000301.BIN
GOTO END

#BIGSETS
SET APPNAME 4624_000301.BIN
GOTO END

#SMALLSETS
SET APPNAME 4606_000301.BIN

#END
GET 46XXSETTINGS.SCR
```

QoS

The 4600 Series IP Telephones support both IEEE 802.1D/Q and DiffServ, and may in the future support other, possibly proprietary, procedures for implementing Quality of Service. In addition, other network-based QoS initiatives such as UDP port selection do not require support by the telephones, but nonetheless can contribute to improved QoS for the entire network.

IEEE 802.1D and 802.1Q

IEEE's 802.1Q standard defines a tag that can be added to voice and data packets. Most of the information associated with this tag deals with Virtual LAN (VLAN) management, but 3 bits are reserved for identifying packet priority. These 3 bits allow any one of 8 priorities to be assigned to a specific packet. As defined in the standard, the 8 priorities are, from highest to lowest:

- 7: Network management traffic
- 6: Voice traffic with less than 10ms latency
- 5: Voice traffic with less than 100ms latency
- 4: "Controlled-load" traffic (mission-critical data applications)
- 3: Traffic meriting "extra-effort" by the network for prompt delivery (for example, executives' e-mail)
- 2: Reserved for future use
- 0: Traffic meriting the network's "best-effort" for prompt delivery (the default priority)

- 1: Background traffic such as bulk data transfers and backups

NOTE Priority 0 is a higher priority than Priority 1.

To support IEEE 802.1D/Q, the 4600 Series IP Telephones can be administered either from the network, via appropriate administration of the DHCP or TFTP servers, or at the telephone itself, via dialpad input. Specific implementation details for local administration are in the *Installation Manual*, and for remote administration are in this chapter, in 4600 Series IP Telephone Scripts and Application Files on page 4-18. In summary, three IEEE 802.1D/Q QoS parameters in the telephones can be administered. These parameters are:

- L2Q: setting the 802.1Q framing parameter ON or OFF
- L2QAUD: setting the 802.1Q audio priority value (between 0 and 7, default is 6)
- L2QSIG: setting the 802.1Q signaling priority value (between 0 and 7, default is 6)

The 4600 Series IP Telephones can simultaneously support receipt of packets using, or not using, 802.1Q parameters.

WARNING:

If you turn VLAN framing on (L2Q=1) using the TFTP-based upgrade script (46xxupgrade.scr), and you set your VLAN ID (L2QVLAN) to something other than your default VLAN ID, the IP address you are using for the telephone will probably be wrong for the new VLAN. See ***Administering Options for the 4600 Series IP Telephones*** *for more information.*

DIFFSERV

As defined in IETF RFCs 2474 and 2475, “services” are basically ways of treating different subsets of a network’s traffic in different ways at the Internet Protocol (IP) layer, Layer 3. For example, some packets might be routed in such a way as to expedite delivery (minimize delay), while others are routed to minimize loss, minimize cost, etc. The differentiation between these services (that is, Differentiated Services) is provided by a redefinition of an octet in the Layer 3 headers for IP versions 4 and 6, also termed IPv4 and IPv6, respectively. This octet is called a Type of Service (TOS) octet in IPv4 and a Traffic Class octet in IPv6, but in both cases the octet is interpreted differently than it was originally defined. With Differentiated Services, bits 0 through 5 of the octet identify a Differentiated Services Code Point (DSCP) that identifies a procedure to be used to handle that packet on a per-hop basis. Bits 6 and 7 of the octet are currently unused, and are ignored by DSCP-compliant routers.

With DiffServ, the default DSCP is all zeroes, and represents “no special handling”. RFC 2474 also defines eight “Class Selector Codepoints”, which are the eight DSCP encodings that can be represented by xxx000 where “x” represents one bit. These Code Selector Codepoints are considered prioritized, with the larger numerical values having a higher relative order. DSCP-compliant routers should treat larger-valued DSCPs in such a way as to give the associated packets a “probability of timely forwarding” greater than a packet with a lower-valued DSCP. In addition to the eight Class Selector Codepoints, a network may define its own DSCPs by defining

encodings that do not terminate in 000. The specific treatment intended by these custom DSCPs will not necessarily be carried out by routers outside the customer's own network.

To support DiffServ, the 4600 Series IP Telephones are administered at the switch in accordance with the documentation for the switch.

UDP Port Selection

Some data networks include equipment that can perform UDP port selection. This is a mechanism by which packets with port numbers in a given range are given priority over packets with port numbers outside that range.

To support UDP port selection, the 4600 Series IP Telephones can be administered either from the network, via appropriate administration of the DHCP or TFTP servers, or at the telephone itself, via dialpad input. Specific implementation details for local administration are in the *4600 Series IP Telephone Installation Guide*, and, for remote administration, in 4600 Series IP Telephone Scripts and Application Files on page 4-18. In summary, the system value MCPORT represents the port on the TN2302AP board. This port number can be used to administer routers, etc. that support UDP port selection, to maximize the priority given to the voice packets being exchanged between the PBX and the telephone.

The default value for MCPORT is 1719. The switch must be administered to use a port within the proper range for the specific LAN, and the IP Telephone(s) will copy that port. A related parameter is PORTAUD, which is the RTP port used by the switch. In accordance with standards RFC 1889 and 1890, the IP Telephone uses a default value for PORTAUD of 5004. MCPORT and PORTAUD are both administrable (see Administering Options for the 4600 Series IP Telephones on page 4-24).

QoS with 4620 and 4630 IP Telephones

As of Release 1.7, both the 4620 and 4630 telephones offer the end user an opportunity to monitor network audio performance while on a call. The user guides for each phone go into specific details on how the user gets to the appropriate screen; this document tells you what the end user can see, and what it means. Specifically, the following parameters are displayed in real-time to users on the appropriate screens, while on a call:

Parameter	Possible Values
Audio Connection Present?	*Yes* (if a receive RTP stream has been established) *No* (if a receive RTP stream has **not** been established)
Received Audio Coding	*G.711 or G.729*
Silence Suppression	*Yes* (if the telephone knows the far-end has silence suppression Enabled) *No* (if the telephone knows the far-end has silence suppression Disabled, or the telephone does not know either way)

Parameter	Possible Values
Packet Loss	"No data" or a decimal percentage. Late and out-of-sequence packets are counted as lost, if they are discarded. Packets are not counted as lost until a subsequent packet is received and the loss confirmed by the RTP sequence number.
Packetization Delay	"No data" or an integer number of milliseconds. The number reflects the amount of delay in received audio packets, and includes any look-ahead delay associated with the codec.
One-way Network Delay	"No data" or an integer number of milliseconds. The number is one-half the value RTCP computes for the round-trip delay.
Network Jitter Compensation Delay	An integer number of milliseconds reporting the average delay introduced by the telephone's jitter buffer.

In addition, based on the current values for the above parameters, the Network Audio Quality Screen presents the user with a qualitative assessment of the overall audio quality currently being experienced. This assessment is based on separate evaluations of the Packet Loss and the total Network Delay (the sum of Packetization Delay, One-way Network Delay, and Network Jitter Compensation Delay), and consideration of the codec in use.

The implication of this information for LAN administration depends, of course, on the values reported by the user and the specific nature of your LAN (topology, loading, QoS administration, etc.) The major use for this information is to give the user an idea of how network conditions are affecting the audio quality of the current call. It is assumed you have more detailed tools available for troubleshooting the LAN.

RSVP and RTCP

Resource ReSerVation Protocol (RSVP) is an IETF-standard protocol used by hosts to request resource reservations throughout a network. RSVP-compliant hosts send messages through a network to receivers, which respond with messages requesting a type of service and an amount of resources (e.g., bandwidth) to carry out that service. The host is responsible for *admitting* (approving) or *rejecting* (denying) the request. In a QoS context, RSVP is used to try to reserve bandwidth in the network for voice calls, on a call-by-call basis. If insufficient bandwidth is available for the target voice quality, a request to use network bandwidth for a voice call will be rejected.

RTP Control Protocol (RTCP), as its name implies, is a protocol that provides control functions for Real-time Transport Protocol (RTP). RTP provides end-to-end network services for real-time data (such as Voice over IP), but does not provide a reservation function, nor does it guarantee any level of QoS. RTCP supplements RTP by monitoring the quality of the RTP services and can provide real-time information to users of an RTP service. In a QoS context, RTCP is valuable for identifying information such as packet loss, 1-way delay (how long a packet has to go from source

A to destination B), jitter, etc. RTCP itself does not improve QoS, but it provides information to you to help identify where problem areas might be.

You cannot change the telephone's RSVP or RTCP parameters directly on the telephone or via TFTP or DHCP administration. The only way to change these parameters is by appropriate administration of the switch. See your DEFINITY®/MultiVantage™ administration material for more detail.

Administering Options for the 4600 Series IP Telephones

As indicated in the Introduction of this chapter, there are many parameters that can be administered for the 4600 Series IP Telephones. This section explains how to change parameters via the DHCP or TFTP servers. In all cases, you will be setting a system parameter in the telephone to a desired value. Table 4-4 lists the parameter names, their default values, the valid ranges for those values, and a description of each one. For DHCP, the parameters below are set as desired to desired values in the DHCP Option as discussed in DHCP Generic Setup earlier in this chapter. For TFTP, the parameters below are set as desired to desired values in the TFTP Script File as discussed in Contents of the Upgrade Script also earlier in this chapter.

TFTP Scripts are the recommended way to administer options on the 4600 Series IP Telephones. Some DHCP applications have limits on the amount of user-specified information; such limits could be exceeded by the administration required, for example, a 4630 with all applications administered.

DNS Addressing

As of Release 1.5, the 4600 IP Telephones support DNS addresses as well as dotted decimal addresses. The telephone attempts to resolve a non-ASCII-encoded dotted decimal IP address by checking the contents of DHCP Option 6, as indicated in DHCP Generic Setup, page 4-7. At least one address in Option 6 must be a valid non-zero dotted decimal address - otherwise, DNS will fail. The text string in the system parameter **DOMAIN** (Option 15, see Table 4-4) is appended to the address(es) in Option 6 before the telephone attempts to resolve the DNS address. If Option 6 contains a list of DNS addresses, they are queried in the order given, if no response is received from previous addresses on the list. Alternatively to administering DNS via DHCP, you may specify the DNS server and/or Domain name in the TFTP script file (in which case, you should SET the values for DNSSRVR and DOMAIN first; then you may use those names later in the script).

NOTE If Options 6 and 15 are appropriately administered with DNS servers and Domain names, respectively, MCIPADD and TFTPSRVR settings need not be specified in the Site Specific Option string.

Customizing the Site-Specific Option Number (SSON)

As discussed in DHCP Generic Setup on page 4-7, the SSON, defined to be 176 for the 4600 Series IP Telephones, can be set to a string. For each system parameter listed in Table 4-4 that you want to include, append the following to the SSON string:

a comma followed by **`name=value`**

where **`name`** is a parameter name and **`value`** is its associated value. Invalid values will cause the data to be ignored for that **`name`**. Customizing the SSON will affect all telephones associated with that DHCP server.

Entering Options via the Telephone Dialpad

Chapter 3 of the *4600 Series IP Telephone Installation Guide* details how to use the local administrative options.

To customize any or all of the QoS parameters locally, follow the “QoS Option Setting” procedure in Chapter 3 of the *4600 Series IP Telephone Installation Guide*.

To enable or disable the secondary Ethernet hub locally, follow the “Secondary Ethernet (Hub) Interface Enable/Disable” procedure in Chapter 3 of the *4600 Series IP Telephone Installation Guide*.

To view the 4600 IP Telephone system parameters, refer to Chapter 5, The View Administrative Option, page 5-8.

Table 4-4. 4600 Series IP Telephone Customizable System Parameters

Parameter Name	Default Value	Description and Value Range
DNSSRVR	" " (Null)	Text string containing the IP address of one or more DNS servers (at least one of which must be a valid, non-zero, dotted decimal address.
DOMAIN	" " (Null)	Text string containing the domain name to be used when DNS names in system values are resolved into IP addresses.
IRSTAT	1	Text string containing status of Infrared interface (0= off/disabled, 1= on/enabled).
L2Q	0	802.1Q framing (1= on or 0= off).
L2QAUD	6	Layer 2 audio priority value (0 to 7).
L2QSIG	6	Layer 2 audio priority value (0 to 7).

Table 4-4. 4600 Series IP Telephone Customizable System Parameters —*Continued*

Parameter Name	Default Value	Description and Value Range
L2QVLAN	0	802.1Q VLAN IDentifier (0 to 4095).
MCPORT	1719	Call server transport-layer port number (0-65535).
PHY2STAT	1	Secondary Ethernet interface status (0=off/disabled, 1=on/enabled). This parameter was called "HUBSTAT" in telephones supporting Releases 1.0 through 1.5; 4600 Series IP Telephones supporting Release 1.6 will respond to changes to "HUBSTAT" by changing "PHY2STAT".

NOTE Table 4-4 applies to all 4600 Series IP Telephones. The 4630 and 4620 IP Telephones have additional, optional administration. See Customizing the 4630 IP Telephone and Customizing the 4620 IP Telephone, below, for more information.

Customizing the 4630 IP Telephone

The 4630 IP Telephone has some unique and powerful capabilities that take advantage of its large display and access to LAN facilities. If your organization has an LDAP-compliant directory on your LAN, or if your organization has a corporate website suitable for displaying on the 4630's 1/4-VGA display, you will need to provide the telephone with key information about the servers that provide those facilities. Specifically, to administer the 4630 telephone for the LDAP Directory application, you must provide the information called for in Table 4-5, below. To administer the 4630 telephone for the Web Access application, you must provide the information called for in Table 4-5 on page 4-27. This information must be provided in a customized script file, in accordance with 4600 Series IP Telephone Scripts and Application Files on page 4-18.

CAUTION:

*For the 4630 to work properly, you must have a 46xxsettings.scr file in the same directory as the 4630 application file. If you do not edit the 46xxsettings.scr file, the 4630 will use default settings only. The 46xxsettings.scr file is part of the *zip file on the software download website, and is also available as a standalone download. If you already have such a file because you downloaded it for a previous release of the 4630, installing the *zip file will overwrite the original file.*

In Table 4-5, parameters shown with a *Mandatory* status must be accurate and non-null for the application to work (however, the Avaya Help website will always be available). Parameters with an *Optional* status may be changed to suit your environment; if they are not changed, the defaults will be used.

Table 4-5. 4630 IP Telephone Customizable System Parameters by Application

Parameter Name	Default Value	Status	Description and Value Range
Phone Application Parameters:			
PHNEMERGNUM	" " (Null)	Optional	Text string of a phone number to be dialed in case of an emergency (e.g., 911)
Directory Application Parameters:			
DIRSRVR	" " (Null)	Mandatory	Text string of dotted decimal IP address, or DNS name, of the server containing the LDAP directory.
DIRTOPDN	" " (Null)	Mandatory	"Directory Topmost Distinguished Name"; text string of the root entry of the LDAP directory.
DIRFULLNAME	cn	Optional	Text string for the customer-specific label for the database field.
DIRTELNUM	telephoneNumber	Optional	Text string for the customer-specific label for the database field containing telephone numbers. The default is the standard LDAP value.
DIRSRCHTIME	0	Optional	Text string for an integer number of seconds; the maximum duration the LDAP directory should spend searching before reporting completion or failure of the search. The default is LDAP-standard for "unlimited duration."

Parameter Name	Default Value	Status	Description and Value Range
DIRUSERID	" " (Null)	Optional	Text string for "Directory User ID," which should be modified if the LDAP server requires a User ID and password for access to the Directory.
DIRSRVRPWD	" " (Null)	Optional	Text string for "Directory Server Password," which should be modified if the LDAP server requires a User ID and password for access to the Directory.
DIRCODING	Latin 1	Optional	Text string identifying the character set used by the LDAP directory. Besides the default value, "ASCII" is the other valid value.
DIRLDAPPORT	389	Optional	Directory LDAP Port; the port used to exchange LDAP messages with the server.
Stock Ticker Application Parameters:			
STKSTAT	1	Optional	Text string identifying whether the 4630 phones are allowed to have the Stock Ticker Application. "1" is the default in the 4630; "0" disables the Stock Ticker Application.
Voice Mail Application Parameters:			
VMLCODING	ASCII	Optional	Text string identifying the character set used by the Voice Mail Server Application. Besides the default value, other supported values include "Latin 1" and "Unicode". Note that "Unicode" includes support only for the Latin characters of Unicode.

Parameter Name	Default Value	Status	Description and Value Range
VMLHOME	" " (Null)	Mandatory	Text string containing the URL of the home page for the Voice Mail Application.
Web Access Application Parameters:			
WEBHOME	" " (Null)	Mandatory	Text string containing the URL of the home page for the Web Access application.
WEBPROXY	" " (Null)	Optional	Text string containing the IP address, in dotted decimal or DNS format, of an HTTP proxy server. This parameter is optional if the web pages to be accessed by the user are all on your organization's intranet.
WEBEXCEPT	" " (Null)	Optional	Text string containing a list of one or more HTTP proxy server exception domains, separated by commas, up to a total of 127 ASCII characters. This parameter is optional if the web pages to be accessed by the user are all on your organization's intranet. If WEBPROXY is null, the value of this parameter is ignored.
WEBPORT	80	Optional	Text string containing the TCP port number for the HTTP proxy server. The default is the TCP default for HTTP. This parameter is optional if the web pages to be accessed by the user are all on your organization's intranet. If WEBPROXY is null, the value of this parameter is ignored.

Parameter Name	Default Value	Status	Description and Value Range
WEBCODING	ASCII	Optional	Text string containing the coding to be used for characters entered in forms on web pages. Besides the default, other supported values include "Latin 1" and "Unicode". Note that "Unicode" includes support only for the Latin characters of Unicode.
Backup/Restore Parameters:			
FTPSRVR	" " (Null)	Mandatory (but can be user-specified	Text string containing a 4-octet for the FTP server to be used for storage and retrieval of 4630 user information (see 4630 Backup/Restore, page 4-
FTPDIR	" " (Null)	Optional (and can be user-specified)	Path name for the directory on FTPSRVR used for storage and retrieval of 4630 user information.

NOTE For assistance in developing local websites tailored to the 4630 IP Telephone's display, see Appendix B, Creating Websites for the 4630 IP Telephone.

4630 Backup/Restore

Backup/Restore automatically saves a phone's speed dial button labels and options/parameter settings. When the Automatic Backup option on a 4630 IP Telephone is set to "Yes" and the FTPSRVR is specified, the FTPSTOR command attempts to save all Speed Dial contents and all system options and (non-password) parameters to the FTP server (specified by FTPSRVR in the directory path specified by FTPDIR). Data is saved in an ASCII text file called "4630data.txt". The System Administrator may optionally specify FTPSRVR and/or FPTDIR via network administration, however, these values may also be specified by the phone's user, as covered in "Chapter 8" of the *4630 IP Telephone User's Guide*. Automatic backup occurs whenever the user executes a Save command on a Speed Dial or Options/Parameter screen.

In addition to Speed Dial labels and associated phone numbers, the following options and non-password parameters are saved during a backup:

Setting/Parameter Name	Type
Idle Timeout	Option
Keyboard Layout	Option
Click Feedback	Option
Edit Dialing	Option
Personalized Ring	Option
Redial	Option
Go to Phone on Answer	Option
Go to Phone on Originate	Option
Call Timer	Option
Alphabetize Entries?	Option
Call Log Active?	Option
Call Log Automatic Archive	Option
Directory User ID	Parameter
Automatic Backup	Option
FTP Server IP Address	Parameter
FTP Directory Path	Parameter
FTP User ID	Parameter
Stock Ticker Active?	Option
STK.*mm*	Parameter
Stock Index DJIA	Option
Stock Index S&P 500	Option
Stock Index Nasdaq	Option

Setting/Parameter Name	Type
Stock Change	Option
Stock Volume	Option

If the Automatic Backup option is set to "No," speed dial data, options and parameter settings are not saved unless the user forces a one-time backup via the appropriate option (as covered in "Chapter 8" of the *4630 IP Telephone User's Guide)*. Restoring backed-up data is done via a separate user option, also covered in "Chapter 8" of the *4630 IP Telephone User's Guide*.

NOTE For specific error messages relating to Backup/Restore, see Table 5-4 on page 5-13.

Call Log Archive

Automatic archiving of the Call Log occurs when:

- The Call Log Automatic Archive option setting is "Yes" and
- The Call Log is more than 50% filled with unarchived entries, or a new (unarchived) log entry occurs within two hours of the last archive (whichever occurs first).

Call Log Archive automatically saves applicable Call Log entries. When the Call Log Automatic Archive option on a 4630 IP Telephone is set to "Yes" and the FTPSRVR is specified, the FTP APPE command attempts to save all call log contents to the FTP server (specified by FTPSRVR in the directory path specified by FTPDIR). Data is saved in a file called "4630calllog.txt". The System Administrator may optionally specify FTPSRVR and/or FPTDIR via network administration, however, these values may also be specified by the phone's user, as covered in "Chapter 8" of the *4630 IP Telephone User's Guide*. Automatic backup occurs whenever the user executes a Save command on a Speed Dial or Options/Parameter screen.

NOTE For specific error messages relating to Archiving, see Table 5-4 on page 5-13.

Customizing the 4620 IP Telephone

In Table 4-6, parameters shown with a *Mandatory* status must be accurate and non-null for the application to work. Parameters with an *Optional* status may be changed to suit your environment; if they are not changed, the defaults will be used.

Table 4-6. 4620 IP Telephone Customizable System Parameters

Parameter Name	Default Value	Status	Description and Value Range
Web Access Application Parameters:			
WMLHOME	" " (Null)	Mandatory	Text string containing the URL of the home page for the Web Access application.
WMLPROXY	" " (Null)	Optional	Text string containing the IP address, in dotted decimal or DNS format, of an HTTP proxy server. This parameter is optional if the web pages to be accessed by the user are all on your organization's intranet.
WMLEXCEPT	" " (Null)	Optional	Text string containing a list of one or more HTTP proxy server exception domains, separated by commas, up to a total of 127 ASCII characters. This parameter is optional if the web pages to be accessed by the user are all on your organization's intranet. If WMLPROXY is null, the value of this parameter is ignored.
WMLPORT	80	Optional	Text string containing the TCP port number for the HTTP proxy server. The default is the TCP default for HTTP. This parameter is optional if the web pages to be accessed by the user are all on your organization's intranet. If WMLPROXY is null, the value of this parameter is ignored.

Parameter Name	Default Value	Status	Description and Value Range
WMLCODING	ASCII	Optional	Text string containing the coding to be used for characters entered in forms on web pages. Besides the default, other supported values include "Latin 1" and "Unicode". Note that "Unicode" includes support only for the Latin characters of Unicode.

NOTE For assistance in developing local websites tailored to the 4620 IP Telephone's display, see Appendix C, Creating Websites for the 4620 IP Telephone.

Troubleshooting Guidelines

5

Introduction

This chapter describes problems that may occur during operation of the 4600 Series IP Telephones and possible ways of resolving these problems.

Error Conditions

Table 5-1 below identifies some of the possible operational problems that might be encountered after successful installation of a 4600 Series IP Telephone. Problems that might be encountered during installation, and how to conduct a Self-Test of the telephone, are discussed in the Troubleshooting chapter of the *4600 Series IP Telephone Installation Guide*. The *4630 IP Telephone User's Guide* and *4620 IP Telephone User's Guide* also contain guidance for users having problems with specific 4630 and 4620 applications, respectively.

NOTE Most of the problems reported by users of a 4600 Series IP Telephone are likely not to be problems with the telephone itself. More likely, the problems will be centered on the LAN, where Quality of Service, server administration, and other issues can impact end-user perception of IP Telephone performance.

Table 5-1. Some Error Conditions in Operation of 4600 Series IP Telephones

Condition	Cause/Resolution
The telephone continually reboots, or reboots continuously about every 15 minutes.	**CAUSE:** This is a firmware fault; the MAC address in memory is corrupted. **RESOLUTION:** The telephone must be returned to Avaya for repair.
The message light on the telephone turns on and off intermittently, but the telephone never registers.	**CAUSE:** This is a hardware fault. **RESOLUTION:** The telephone must be returned to Avaya for repair.

Table 5-1. Some Error Conditions in Operation of 4600 Series IP Telephones —*Continued*

Condition		Cause/Resolution
The telephone stops working in the middle of a call,	**AND** No lights are lit on the phone and the display is not lit.	**CAUSE:** Loss of power **RESOLUTION:** Check the connections between the telephone, the power supply, and the power jack.
The telephone stops working in the middle of a call,	**AND** Power to the telephone is fine (and the telephone may have gone through the restarting sequence).	**CAUSE:** DHCP Lease expired or DHCP server not available when telephone attempts to renegotiate DHCP lease. **RESOLUTION:** Short-term, restart the telephone. Long-term, administer the DHCP application to allow longer leases ("Infinite" or "Unlimited" is recommended).
The telephone had been working, but does not work now,	**AND** no lights are lit on the phone and the display is not lit.	**CAUSE:** Loss of power. **RESOLUTION:** Check the connections between the telephone, the power supply, and the power jack.
	power to the telephone is fine, but there is no dialtone. (Display might show "System Busy").	**CAUSE:** Loss of communication with the PBX switch **RESOLUTION:** Check LAN continuity from the PBX to the telephone (using ping or trace-route), and from the telephone to the PBX (by invoking a feature button). Verify the LAN administration has not changed for the TN799 or TN 2302AP boards, or the LAN equipment (routers, servers, etc.) between the switch and the telephone. Verify no one has locally changed the telephone settings (by using the VIEW and ADDR codes, as described in the *4600 Series IP Telephone Installation Guide*). Verify the volume on the telephone is set high enough. Finally, conduct a self-test.

Table 5-1. Some Error Conditions in Operation of 4600 Series IP Telephones —*Continued*

Condition		Cause/Resolution
The telephone had been working, but does not work now, (continued)	**AND** the telephone was recently moved.	**CAUSE:** Loss of communication with the PBX. **RESOLUTION:** As above, but pay particular attention to the possibility that the telephone is being routed to a different DHCP server, or even a different PBX switch. If so, the new server or switch may need to be administered to support the telephone.
	the network was recently changed (servers upgraded or replaced, your DEFINITY/MultiVantage switches, etc.).	**CAUSE:** Loss of communication with the PBX. **RESOLUTION:** As above.
The telephone works, but the audio quality is poor, specifically:	the user hears echo when speaking on a handset.	**CAUSE:** Echo from digital-to-analog conversion on your DEFINITY/ MultiVantage trunk. **RESOLUTION:** Verify which trunk is causing the echo, and swap the trunk's Trunk Termination parameter on the PBX.
	the user hears echo on a headset, but not on a handset.	**CAUSE:** Improper headset adapter. **RESOLUTION:** Replace adapter with Avaya's M12LU or 3412-HIC adapters. The M12LU is recommended, since it supports Automatic Gain Control.
	the user is on speakerphone and hears no echo, but the far end hears echo.	**CAUSE:** Room acoustics **RESOLUTION:** Ensure there are six inches or so of blank space to the right of the telephone. If that is insufficient, use the handset.

Table 5-1. Some Error Conditions in Operation of 4600 Series IP Telephones —*Continued*

Condition		Cause/Resolution
	the user experiences sudden silences (gaps in speech), static, clipped or garbled speech, etc.	**CAUSE:** Jitter, delay, dropped packets, etc. **RESOLUTION:** One or more Quality of Service (QoS) features should be implemented in the network (See Chapter 3 of the *4600 Series IP Telephone Installation Guide*). **CAUSE:** Improper (non-Category 5) wiring. **RESOLUTION:** Replace non-Category 5 wiring with Category 5 wiring.
The 4612 or 4624 IP Telephone works properly except the phone does not ring.		**CAUSE:** The Ringer Off (**RngOF**) softkey feature has been activated. **RESOLUTION:** Use the softkey **Menu** option to access the **RngOF** feature. A downward-pointing triangle means the Ringer is off. Ensure the triangle points up. Also, check the Volume setting on the telephone. Finally, do a Self-test on the telephone.
The telephone works properly except for the speaker.		**CAUSE:** The Speaker was turned off on the PBX. **RESOLUTION:** Administer the PBX to allow that station's speaker to operate. If that does not work, do a Self-test on the telephone.
The telephone works properly, except incoming DTMF tones are not received.		**CAUSE:** The TN2302AP board does not pass in-band DTMF tones. **RESOLUTION:** None; the board is operating as designed.

Table 5-1. Some Error Conditions in Operation of 4600 Series IP Telephones —*Continued*

Condition	Cause/Resolution
The telephone works properly, except sidetone DTMF is not heard.	**CAUSE:** PBX suppresses sidetone DTMF. **RESOLUTION:** On PBX administration, on the Change-System-Parameters screen, enable On-Hook Dialing. If the user has Hands-Free Answer (HFA) and answers the call on the telephone's speakerphone, then switches to the handset, pressing the dialpad buttons does not send DTMF tones. This is a known bug, and the only current resolution is to disable HFA.
Hands-Free Answer (HFA) is administered but the telephone did not automatically answer a call.	**CAUSE:** HFA only works if the telephone is idle. If a second call comes into the telephone while the first call is in progress (including ringing before the first call is answered), the second call is ignored. **RESOLUTION:** None.
The TFTP application terminates and asks for registration.	**CAUSE:** Non-Avaya shareware or freeware TFTP applications often cease operating to request registration. **RESOLUTION:** *Short-term:* Restart the application. *Long-term:* Register the product or replace it with an application that does not behave this way (for example, Avaya's TFTP application).

Table 5-1. Some Error Conditions in Operation of 4600 Series IP Telephones —*Continued*

Condition		Cause/Resolution
The TFTP script file is ignored or not used by the telephone		**CAUSE:** The telephone expects lines of the script file to terminate with a <Carriage Return><Line Feed>. Some UNIX applications only terminate lines with <Line Feed>. Editing the script file with a UNIX-based editor can strip <Carriage Return>s from the file, causing the entire file to be treated as a comment, and thus be ignored. **RESOLUTION:** Edit the script file with a Windows-based editor, or another editor that does not strip out the <Carriage Return>.
	AND the TFTP server is a LINUX or UNIX system.	**CAUSE:** UNIX and LINUX systems use case-sensitive addressing and file labels. **RESOLUTION:** Verify the file names and path in the script file are accurately specified.
Power to the telephone is interrupted while the telephone is saving the application file **and** the TFTP application hangs.		**CAUSE:** The TFTP server hangs if power is interrupted while a telephone is saving the application file. **RESOLUTION:** Restart the TFTP server.
The DHCP server indicates the 4600 Telephone reports itself as a Token Ring device, and refuses to provide the telephone an address.		**CAUSE:** Early versions of the 4600 Series Telephones erroneously report being a Token Ring device. With most DHCP servers, this does not matter. Some LINUX servers, however, will refuse to issue addresses to Ethernet devices reporting to be Token Ring devices. **RESOLUTION:** Administer the DHCP server to delete all MAC and IP addresses associated with Lucent Technologies or Avaya, or allow the associated DHCP leases to expire.

Table 5-1. Some Error Conditions in Operation of 4600 Series IP Telephones —*Continued*

Condition	Cause/Resolution
The user indicates a 4620-specific or 4630-specific application is not accessible.	**CAUSE:** The 46xxsettings script file is not pointed to accurately, or is not properly administered to allow the application. **RESOLUTION:** Assuming the user is meant to have that application, verify the 46xxsettings script file is properly specified for your system, including case (if your TFTP server is UNIX or LINUX) and extension. Then, verify all the relevant parameters (as indicated in Table 4-5 and Table 4-6, as appropriate) are accurately specified in the 46xxsettings file.

There are three areas where installers can troubleshoot problems before seeking assistance from the system or LAN administrator:

1. Check the wiring (power and Ethernet) for the following:
   - Whether all components are plugged in correctly.
   - Check LAN connectivity in both directions to all servers (DHCP, TFTP, Call Server)
   - If the telephone is supposed to be powered from the LAN, ensure the LAN is properly administered and is compliant with IEEE 802.3af.
2. If you are using static addressing, do the following:
   - Use the View command to find the names of the files being used and verify that these filenames match those on the TFTP server. Check on the Avaya website to verify whether the correct files are being used.
   - Use the ADDR option to verify IP addresses.
   - Use the QOS option to verify QoS parameters.

     Refer to Chapter 3 of the *4600 Series IP Telephone Installation Guide*.
3. If the 4600 Series IP Telephone is not communicating with the system (DHCP, TFTP, or Call Server), make a note of the last message that was displayed and consult the system administrator.

The View Administrative Option

If you are using static addressing and encounter problems, use the following procedure to verify the current values of system parameters and file versions.

NOTE Also use the ADDR option to view IP addresses (see "Static Addressing Installation" in Chapter 3 of *4600 Series IP Telephone Installation Guide*). These may have been entered incorrectly*. Verify whether you were provided with correct IP addresses.*

1. While the phone is on-hook and idle, press the following sequence of keys on the faceplate of the telephone:

   Mute 8 4 3 9 # (**Mute V I E W #**)

   NOTE Press the **Mute** key momentarily. Do not press this key while pressing other keys. The 4630 IP Telephone does not have a dedicated **Hold** button; for all other 4600 Series IP Telephones, pressing the **Hold** button instead of the **Mute** button will also work.

   The following text is displayed left-justified at the top of the display:

   ```
   View settings
   *=next   #=exit
   ```

2. Press the ***** button at any time during viewing to display the next name and system value pair from the list below, returning to the first pair after the last pair has been displayed.

   Press the # button at any time during viewing to terminate the procedure and restore the user interface to its previous state.

The names and values are displayed in the following order:

Table 5-2. Parameter Values

Name	System Value	Format
Model	**46ccDccc**	up to 8 ASCII graphics characters
Market	**domestic** **export**	Only one value is displayed
Phone SN	**ccccccccccc** **cccccccc**	Phone Serial Number, up to 18 ASCII graphic characters
PWB SN	**ccccccccccc** **cccccccc**	Printed Wiring Board (circuit board) Serial Number, up to 18 ASCII graphic characters
PWB comcode	**ccccccccc**	9 ASCII numbered characters
MAC address	**00:60:1D:hh:hh:hh**	Each octet of the MAC address is displayed as a pair of hexadecimal numbers.
	filename1.exe **filename2.exe**	up to 16 ASCII graphic characters. Out of the box, there will be only one **filename.exe**. After installation, there should be two filenames. If there is only one, installation has failed. Verify the problem, and then re-install the telephone.

Error Messages

The 4600 Series IP Telephones issue messages in English only. The IP Telephones also display messages from the switch, which outside the United States may issue messages in the local language.

Table 5-3. Possible Error Messages During Installation or Operation of 4600 Series IP Telephones

Error Message	Cause/Resolution
During Installation	
Checksum error	**CAUSE:** Downloaded application file was not downloaded or saved correctly. **RESOLUTION:** The telephone automatically resets and attempts to re-initialize.

Table 5-3. Possible Error Messages During Installation or Operation of 4600 Series IP Telephones—*Continued*

Error Message	Cause/Resolution
DHCP: CONFLICT * to program	**CAUSE:** At least one of the IP address offered by the DHCP server conflicts with another address. **RESOLUTION:** Review DHCP server administration to identify duplicate IP address(es)
File too large Cannot save file	**CAUSE:** The telephone does not have sufficient room to store the downloaded file. **RESOLUTION:** Verify the proper filename is administered in the TFTP script file, and that the proper application file is located in the appropriate location on the TFTP server.
Gateway Error	**CAUSE:** DEFINITY Release 8.4 does not have an H.323 station extension for this telephone. **RESOLUTION:** On the station administration screen, ensure the DCP set being aliased for this IP telephone has an H.323 station extension administered, in accordance with switch administration instructions.
Hardware failure	**CAUSE:** Hardware failure prevented downloading of application file, **RESOLUTION:** Replace telephone.
IP Address in use by another	**CAUSE:** The telephone has detected an IP address conflict. **RESOLUTION:** Verify administration to identify duplicate IP address(es).
No Ethernet	**CAUSE:** When first plugged in, the IP Telephone is unable to communicate with the Ethernet. **RESOLUTION:** Verify the connection to the Ethernet jack, verify the jack is Category 5, verify power is applied on the LAN to that jack, etc.
No file server address	**CAUSE:** The TFTP server IP address in the IP telephone's memory is all zeroes. **RESOLUTION:** Depending on the specific requirements of your network, this may not be an error. If appropriate, either administer the DHCP server with the proper address of the TFTP server, or administer the telephone locally using the ADDR option as detailed in the *4600 Series IP Telephone Installation Guide*.

Table 5-3. Possible Error Messages During Installation or Operation of 4600 Series IP Telephones—*Continued*

Error Message	Cause/Resolution
System busy	**CAUSE:** Most likely, the number of IP endpoints on the DEFINITY/MultiVantage is already at maximum, Less likely, network resource is unavailable. **RESOLUTION:** The telephone was attempting to access a network resource (DHCP server, TFTP server, or the DEFINITY/ MultiVantage) and was not successful. The resource being called upon should be checked for its availability. If it appears operational and properly linked to the network, verify addressing is accurate and a communication path exists in both directions between the telephone and the resource.
System Error	**CAUSE:** DEFINITY/MultiVantage has an unspecified problem. **RESOLUTION:** Consult your DEFINITY or MultiVantage Administration and Troubleshooting documentation.
During Registration	
Extension error	**CAUSE:** An invalid DEFINITY/MultiVantage Registration extension has been entered. **RESOLUTION:** Re-enter the extension if mis-entered initially. If appropriate, verify proper extension with respect to switch administration.
Extension in use	**CAUSE:** The specified extension is already in use, according to the PBX switch. **RESOLUTION:** Wait a few minutes, and try again. This will work for the case when the extension is correctly administered, but service was interrupted and DEFINITY is not yet aware of that fact. Otherwise, verify proper extension with respect to switch administration.
Failed to set phone IP address	**CAUSE:** The 4600 Series Telephone was originally installed on one switch with Static Addressing, and has subsequently been installed on another switch with an active DHCP server assigning dynamic IP addresses. **RESOLUTION:** Reset the telephone.
Incompatible	**CAUSE:** This release of DEFINITY does not support the current version of the IP Telephone. **RESOLUTION:** Upgrade to the current version of MultiVantage software.

Table 5-3. Possible Error Messages During Installation or Operation of 4600 Series IP Telephones—*Continued*

Error Message	Cause/Resolution
Message light blinks on and off, and the telephone did not complete registration.	**CAUSE:** The telephone has a hardware fault. **RESOLUTION:** Replace the telephone.
Password Error	**CAUSE:** An invalid PBX Registration password has been entered. **RESOLUTION:** Re-enter the password if mis-entered initially. If appropriate, verify proper password with respect to switch administration.
Resource Error	**CAUSE:** DEFINITY rejects the registration request. **RESOLUTION:** Verify your DEFINITY or MultiVantage Administration to ensure the telephone's proper IP address, extension, and password are being used.
Timeout Error	**CAUSE:** Protocol timeout error. **RESOLUTION:** Retry. If failure continues, check network congestion, addresses, etc. to identify cause of timeout.
Undefined Error	**CAUSE:** An error has occurred without an identifiable cause. **RESOLUTION:** Conduct Self-test, restart the telephone, and if no other cause becomes evident, replace the telephone.
Wrong Set Type	**CAUSE:** The PBX does not recognize the set type. **RESOLUTION:** Ensure the PBX is properly administered to expect the appropriate telephones for the IP address and extension.

Table 5-4. Possible Error Messages During 4630 Backup/Restore

Error Message	Cause/Resolution
Current options and Speed Dial entries have not yet been backed up.	**CAUSE:** An attempt to save the current options and speed dial entries on the FTP server has not yet been made or is in progress. **RESOLUTION:** Wait for a message stating that backup was successful.
The FTP Server Name is not known. Please check the FTP Server IP Address	**CAUSE:** Invalid or missing Server name. **RESOLUTION:** Verify the FTPSRVR address is the File Server to which backup data should be saved. Verify that the FTPDIR value is the correct directory path for that File Server.
The FTP Server has not yet responded, so backup has not yet succeeded.	**CAUSE:** The FTP Server has not responded to the attempt to backup/restore. **RESOLUTION:** Try again, verify the FTP server address, verify the FTP server is online, and/or verify the network connectivity.
The FTP Server has denied access. Please check FTP Setup parameters.	**CAUSE:** The FTP Server has reported that it did not store data. **RESOLUTION:** Verify the FTP server setup parameters, as indicated on the FTP Setup Parameters screen (see "FTP Setup" Chapter 8 of the *4630 IP Telephone User's Guide*.
The FTP Server was unable to store the backup file.	**CAUSE:** The FTP Server has reported that it could not store the data. **RESOLUTION:** Verify administration and available capacity/ filespace on the FTP Server.

Appendix A: Avaya - 46xx IP Telephone MIB

A

Avaya - 46xx IP Telephone MIB

```
Avaya-46xxIPTelephone-MIB

DEFINITIONS ::= BEGIN

IMPORTS

        enterprises, OBJECT-TYPE, MODULE-IDENTITY, Integer32, IpAddress

                FROM SNMPv2-SMI

                        DisplayString

                        FROM RFC1213-MIB ;


endpointMIB MODULE-IDENTITY

                    LAST-UPDATED    "200206182000Z"

                    ORGANIZATION    "avaya"

                   CONTACT-INFO    "Robert Mitchell

                                email : robmitchell@avaya.com

                            (732) 817-6330"

                    DESCRIPTION     "MIB Module for 46xx IP Telephones through Release 1.61"

            ::= { ipEndpointMIBs 1 }


avaya               OBJECT IDENTIFIER ::= { enterprises 6889 }

avayaProducts   OBJECT IDENTIFIER ::= { avaya 1 }

avayaMibs           OBJECT IDENTIFIER ::= { avaya 2 }

ipEndpointProducts    OBJECT IDENTIFIER ::= { avayaProducts 69 }
```

```
ipEndpointMIBs  OBJECT IDENTIFIER ::= { avayaMibs 69 }

endptID         OBJECT IDENTIFIER ::= { endpointMIB 1 }

endptMARKET     OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "ID designation. This variable returns a string
                   specifying ’domestic’ or ’export’."
                ::= { endptID 1 }
endptMODEL      OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Model identifier. This variable returns a string
                   of up to 8 ASCII characters."
                ::= { endptID 2 }
endptMCIPADD    OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "List of call server addresses.  This variable
                   returns a sequence of IP addresses, in dotted-decimal or
                   DNS format and separated by commas, of call server IP
                   addresses available for use."
                ::= { endptID 3 }
```

```
endptMCIPINUSE    OBJECT-TYPE
                SYNTAX      IpAddress
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Call server address.  This variable returns a
                   4-octet string specifying the IP address of the call
                   server currently in use. "
                ::= { endptID 4 }
endptMCPORT     OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Transport layer port number. This variable
                   returns an integer specifying the port number in use
                   by the current call server."
                ::= { endptID 5 }
endptPHONESN    OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Housing serial number. This variable returns
                   a string of up to 18 ASCII characters, specifying the
                   serial number on the outside of the endpoint's housing."
                ::= { endptID 6 }
```

```
endptPWBCC      OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "PWB comcode. This variable returns a string of up to 9
                   numerical characters, specifying the comcode of the printed wiring
                   board of the endpoint."
                ::= { endptID 7 }
endptPWBSN      OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "PWB serial number. This variable returns a
                   string of up to 18 ASCII characters, specifying the
                   serial number on the endpoint’s printed wiring board."
                ::= { endptID 8 }
endptETHERADD   OBJECT-TYPE
                SYNTAX      OCTET STRING
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "MAC address. This variable returns a 6-octet string
                   specifying the endpoint’s MAC address."
                ::= { endptID 9 }
```

```
endptESPEED     OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Ethernet interface speed. This variable returns
                   a string specifying the endpoint's Ethernet interface
                   speed in Mbps: '100', '10', or '0'."
                ::= { endptID 10 }
endptIPADD      OBJECT-TYPE
                SYNTAX      IpAddress
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "IP address. This variable returns a 4-octet
                   string specifying the endpoint's current IP address."
                ::= { endptID 11 }
endptDHCPLEASETIME      OBJECT-TYPE
                SYNTAX      Integer32
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Duration of IPADD lease. This variable returns
                   an unsigned integer specifying the length of time of the
                   IPADD lease."
                ::= { endptID 12 }
```

```
endptDHCPLEASERENEW     OBJECT-TYPE
                SYNTAX      Integer32
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Renewal time of IPADD lease. This variable
                   returns an unsigned integer specifying the length
                   of time at which to renew the IPADD lease."
                ::= { endptID 13 }
endptDHCPLEASEREBIND     OBJECT-TYPE
                SYNTAX      Integer32
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Rebind time of IPADD lease. This variable returns
                   an unsigned integer specifying the length of time at
                   which to rebind the IPADD lease."
                ::= { endptID 14 }
endptGIPADD     OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "List of router addresses.  This variable returns
                   a sequence of IP addresses, in dotted-decimal format
                   and separated by commas, of routers available for use."
                ::= { endptID 15 }
```

```
endptGIPINUSE       OBJECT-TYPE
                SYNTAX      IpAddress
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Router address.  This variable returns a 4-octet
                   string specifying the IP address of the router currently
                   in use."
                ::= { endptID 16 }
endptNETMASK        OBJECT-TYPE
                SYNTAX      IpAddress
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Subnet mask. This variable returns a 4-octet
                   string specifying the subnet mask currently in use."
                ::= { endptID 17 }
endptTFTPDIR    OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "TFTP server path name. This variable returns
                   a string of 0-127 ASCII characters, specifying
                   the path name currently prepended to file names used in
                   TFTP read operations."
                ::= { endptID 18 }
```

```
endptTFTPSRVR   OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "List of TFTP server addresses.  This variable
           returns a sequence of IP addresses, in dotted-decimal or
           DNS format and separated by commas, of TFTP servers
           available for use."
        ::= { endptID 19 }
endptTFTPINUSE      OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "TFTP server address.  This variable returns a
           4-octet string specifying the IP address of the TFTP
           server currently in use."
        ::= { endptID 20 }
endptBOOTNAME       OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Boot code file ID. This variable returns
           a string specifying the boot code file
           used to initiate telephone operation."
        ::= { endptID 21 }
```

```
endptAPPNAME    OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Application file ID. This variable returns
                   a string specifying the system-specific application file
                   that should be used for telephone operation."
                ::= { endptID 22 }
endptSSON       OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Site-specific option number (SSON). This variable
                   returns an integer between 128 and 254 inclusive,
                   specifying the SSON in use.  The default is 176."
                ::= { endptID 23 }
endptBBURST     OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      obsolete
                DESCRIPTION "Blackburst flag. This variable returns 0
                   if Blackburst is currently OFF, else 1."
                ::= { endptID 24 }
```

```
endptHUBSTAT    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      obsolete
                DESCRIPTION "Hub flag. This variable has been superseded by
                   endptPHY2STAT."
                ::= { endptID 25 }
endptDSCPAUD    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "DSCP for audio. This variable specifies the
                   Differentiated Services Code Point for audio currently
                   in use (between 0 and 63 inclusive)."
                ::= { endptID 26 }
endptDSCPSIG    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION " DSCP for signaling. This variable specifies the
                   Differentiated Services Code Point for signaling
                   currently in use (between 0 and 63 inclusive)."
                ::= { endptID 27 }
```

```
endptL2Q        OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "802.1Q flag. This variable returns 0 if the Layer
                   2 802.1Q framing parameter is currently OFF, else 1."
                ::= { endptID 28 }
endptL2QAUD     OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Audio priority. This variable returns an integer
                   between 0 and 7 inclusive specifying the current Layer 2
                   802.1Q audio priority value."
                ::= { endptID 29 }
endptL2QSIG     OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Signaling priority. This variable returns an
                   INTEGER  between 0 and 7 inclusive specifying the current
                   Layer 2 802.1Q signaling priority value."
                ::= { endptID 30 }
```

```
endptL2QVLAN    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "VLAN Identifier. This variable returns an integer
                   between 0 and 4095 inclusive specifying the current
                   Layer 2 802.1Q VLAN Identifier value."
                ::= { endptID 31 }
endpt46XXUPGR   OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Upgrade script ID. This variable returns
                   a string of up to 16 ASCII characters specifying the
                   name of the upgrade script retrieved by TFTP after the
                   most recent power-up or reset."
                ::= { endptID 32 }
endptDNSSRVR   OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "List of DNS server addresses.  This variable
                   returns a sequence of IP addresses, in dotted-decimal
                   format and separated by commas, of DNS servers
                   available for use."
                ::= { endptID 33 }
```

```
endptDNSINUSE  OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "DNS server address.  This variable returns a
           4-octet string specifying the IP address of the DNS
           server currently in use."
        ::= { endptID 34 }
endptDOMAIN    OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "List of DNS domain labels.  This variable
           returns a sequence of labels, separated by periods, for
           use as the highest-level domain in resolving DNS names
           in system values."
        ::= { endptID 35 }
endptRTCPMON  OBJECT-TYPE
        SYNTAX      IpAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "RTCP Monitor address.  This variable returns a
           4-octet string specifying the IP address of the RTCP
           monitor currently in use."
        ::= { endptID 36 }
```

```
endptPHY2STAT    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Hub flag. This variable returns 0 if the
                   endpoint’s secondary Ethernet interface is currently NOT
                   enabled, else 1."
                ::= { endptID 37 }
endptIRSTAT    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "IR Interface flag. This variable returns 0 if the
                   endpoint’s IR interface is currently NOT enabled, else 1."
                ::= { endptID 38 }
endptSMTPSRVR    OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "SMTP server IP addresses.  This variable returns a
                   comma-separated list of SMTP server IP addresses available
                   for use, in dotted-decimal format."
                ::= { endptID 39 }
```

```
endptDSPVERSION  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "DSP Version.  This variable returns a string of
          up to 10 characters, that identifies the version of the
          DSP firmware in the endpoint."
        ::= { endptID 40 }
endptLOGSRVR   OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Event Log server address.  This variable
          returns at most one IP address, in dotted-decimal
          or DNS format, of the server to be used for event logging.  This
          function is intended for Avaya use, in debugging if necessary."
        ::= { endptID 41 }
endptLOGSTAT    OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Event Log flag. This variable returns 0 if the
          endpoint's event logging capability is currently NOT
          enabled, else 1.  This function is intended for Avaya use,
          in debugging if necessary."
        ::= { endptID 42 }
```

```
endptAGCHAND   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Handset AGC status. This variable returns 1 if the
           endpoint’s handset Automatic Gain Control capability is currently
           enabled, else 0."
        ::= { endptID 43 }
endptAGCHEAD   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Headset AGC status. This variable returns 1 if the
           endpoint’s headset Automatic Gain Control capability is currently
           enabled, else 0."
        ::= { endptID 44 }
endptPHONECC   OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Phone comcode. This variable returns a string of up to 9
           numerical characters, specifying the comcode of the endpoint itself."
        ::= { endptID 45 }


endptNVM       OBJECT IDENTIFIER ::= { endpointMIB 2 }
```

```
endptNVMCIPADD  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "IP address of call server stored in Non-Volatile
           Memory (NVM). This variable returns a 4-octet string or DNS string.
           Default is AvayaCallServer."
        ::= { endptNVM 1 }
endptNVMCPORT   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM Transport layer port number. This variable
           returns an integer specifying the call server port number
           stored in NVM."
        ::= { endptNVM 2 }
endptNVIPADD    OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM IP address. This variable returns a 4-octet
           string specifying the endpoint's IP address stored in
           NVM."
        ::= { endptNVM 3 }
```

```
endptNVGIPADD   OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM Router address.  This variable returns a 4-
           octet string specifying the IP address of the router as
           stored in NVM."
        ::= { endptNVM 4 }
endptNVNETMASK  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM Subnet mask. This variable returns a 4-octet
           string specifying the subnet mask as stored in NVM."
        ::= { endptNVM 5 }
endptNVTFTPSRVR OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM TFTP server address. This variable returns
           a 4-octet string specifying the IP address of the file
           server as stored in NVM.  The default is AvayaTFTPserver."
        ::= { endptNVM 6 }
```

```
endptNVSSON     OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "NVM Site-specific option number. This variable
                   returns an integer between 128 and 254 inclusive,
                   specifying the SSON stored in NVM.  The default is 176."
                ::= { endptNVM 7 }
endptNVBBURST   OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      obsolete
                DESCRIPTION "NVM Blackburst flag. This variable returns 0
                    if the Blackburst parameter in NVM is OFF, else 1."
                ::= { endptNVM 8 }
endptNVHUBSTAT  OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      obsolete
                DESCRIPTION "NVM Hub flag. This variable has been superseded by
                   endptNVPHY2STAT."
                ::= { endptNVM 9 }
```

```
endptNVDSCPAUD  OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM DSCP for audio. This variable specifies the
           Differentiated Services Code Point for audio stored
           in NVM (integer between 0 and 63 inclusive)."
        ::= { endptNVM 10 }
endptNVDSCPSIG  OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION " NVM DSCP for signaling. This variable specifies
           the Differentiated Services Code Point for signaling
           stored in NVM (integer between 0 and 63 inclusive)."
        ::= { endptNVM 11 }
endptNVL2Q      OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM 802.1Q flag. This variable returns 0 if the
           Layer 2 802.1Q framing parameter in NVM is OFF, else 1."
        ::= { endptNVM 12 }
```

```
endptNVL2QAUD   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM Audio priority. This variable returns an
           integer between 0 and 7 inclusive specifying the Layer 2
           802.1Q audio priority value stored in NVM."
        ::= { endptNVM 13 }
endptNVL2QSIG   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM Signaling priority. This variable returns an
           integer between 0 and 7 inclusive specifying the Layer 2
           802.1Q signaling priority value stored in NVM."
        ::= { endptNVM 14 }
endptNVL2QVLAN  OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM VLAN Identifier. This variable returns an
           integer between 0 and 4095 inclusive specifying the Layer
           2 802.1Q VLAN Identifier value stored in NVM."
        ::= { endptNVM 15 }
```

```
endptNVPHY2STAT  OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "NVM Hub flag. This variable returns 0 if the
           endpoint’s secondary Ethernet interface parameter in
           NVM is NOT enabled, else 1."
        ::= { endptNVM 16 }
endptNVLOGSTAT    OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Event Log flag. This variable returns 0 if the
           endpoint’s event logging capability in NVM is currently
           NOT enabled, else 1."
        ::= { endptNVM 17 }
endptNVAGCHAND   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Handset AGC status. This variable returns 1 if the
           endpoint’s handset Automatic Gain Control capability in NVM is
           currently enabled, else 0."
        ::= { endptNVM 18}
```

```
endptNVAGCHEAD   OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION "Headset AGC status. This variable returns 1 if the
               endpoint's headset Automatic Gain Control capability in NVM is
               currently enabled, else 0."
            ::= { endptNVM 19}
endptMaintenance        OBJECT IDENTIFIER ::= { endpointMIB 3 }
endptUPGRADESCRIPT      OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION "Script loading status flag. This variable
               returns '0' if the upgrade script (endpt46XXUPGR)
               was successfully obtained and interpreted since the
               last power-up or reset. This variable returns '1' if
               the upgrade script was successfully obtained and but
               NOT successfully interpreted since the last power-up
               or reset. This variable returns '2' if the upgrade
               script was NOT successfully obtained since the last
               power-up or reset."
            ::= { endptMaintenance 1 }
```

```
endptAPPINUSE    OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Application file name.  This variable returns
                   the name of the application file in the endpoint."
                ::= { endptMaintenance 2 }
endptAPPSTAT    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Application download flag. This variable
                   returns ’0’ if the most recent attempt to download
                   and store an application file was successful. This
                   variable returns ’1’ if the most recent attempt to
                   download an application file was successful but the
                   file was not successfully stored in NVM.  This variable
                   returns ’2’ if the most recent attempt to download an
                   application file was NOT successful."
                   ::= { endptMaintenance 3 }
endptDEFINITY   OBJECT IDENTIFIER ::= { endpointMIB 4 }
```

```
endptPORTAUD    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Transport-layer audio port. This variable returns
                   the endpoint's transport layer port number for audio,
                   currently in use."
                ::= { endptDEFINITY 1 }
endptPORTSIG    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Transport-layer signaling port. This variable
                   returns the endpoint's transport layer port number for
                   signaling, currently in use."
                ::= { endptDEFINITY 2 }
endptFEIPADD    OBJECT-TYPE
                SYNTAX      OCTET STRING
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Audio endpoint address. This variable returns
                   a 4-octet string specifying the address of the far-end
                   audio endpoint currently in use."
                ::= { endptDEFINITY 3 }
```

```
endptFEPORT     OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Audio endpoint port number. This variable returns
                   the far-end endpoint’s transport layer port number for
                   audio, currently in use."
                ::= { endptDEFINITY 4 }
endptCODECR     OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Receive codec. This variable returns the
                   designation of the Receive codec currently in use."
                ::= { endptDEFINITY 5 }
endptCODECT     OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Transmit codec. This variable returns the
                   designation of the Transmit codec currently in use."
                ::= { endptDEFINITY 6 }
```

```
endptJCPC       OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Jitter compensation packet count. This variable
                   returns the Jitter compensation packet count currently
                   in use (between 0 and 20, inclusive)."
                ::= { endptDEFINITY 7 }
endptTMSEC      OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Transmit buffer fill level. This variable returns the
                   transmit buffer minimum fill level, in msec, currently in use."
                ::= { endptDEFINITY 8 }
endptNVPHONEXT      OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Endpoint's phone extension.  This variable
                   returns a text string identifying the endpoint's extension,
                   in free text format.  The default value is NULL. "
                ::= { endptDEFINITY 9 }
```

```
endptL2QBBE     OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "802.1Q BBE priority value. This variable returns
                   an integer between 0 and 7 inclusive specifying the
                   far-end endpoint’s Layer 2 Better than Best Effort audio priority
                   value."
                ::= { endptDEFINITY 10 }
endptDSCPBBE    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "DSCP BBE priority value. This variable returns
                   an integer between 0 and 7 inclusive specifying the
                   far-end endpoint’s DiffServ Better than Best Effort audio priority
                   value."
                ::= { endptDEFINITY 11 }
endptRTCPCONT   OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "RTCP Control flag. This variable returns
                   0 if the RTCP control status is off/disabled, else 1."
                ::= { endptDEFINITY 12 }
```

```
endptRTCPFLOW   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "RTCP Flowrate. This variable returns
           an integer between 0 and 65335 inclusive specifying the
           RTCP flowrate in milliseconds."
        ::= { endptDEFINITY 13 }
endptRSVPCONT   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "RSVP Control flag. This variable returns
           0 if the RSVP control status is off/disabled, else 1."
        ::= { endptDEFINITY 14 }
endptRSVPRFRSH   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "RSVP Refresh rate. This variable returns
           an integer between 0 and 65335 inclusive specifying the
           RSVP refresh rate in milliseconds."
        ::= { endptDEFINITY 15 }
```

```
endptRSVPRTRY   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "RSVP Failed Retry flag. This variable returns
           0 if the RSVP Failed Retry Reservation control status is
           off/disabled, else 1."
        ::= { endptDEFINITY 16 }
endptRSVPPROF   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "RSVP profile flag. This variable returns
           0 if the RSVP profile is guaranteed, else 1 for controlled
           load."
        ::= { endptDEFINITY 17 }

endptAdvApps    OBJECT IDENTIFIER ::= { endpointMIB 5 }
endptCIBURL     OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      obsolete
        DESCRIPTION "Documentation URL. This variable returns
           a URL of the directory in which the web-based Customer
           Information Booklet web pages are located."
        ::= { endptAdvApps 1 }
```

```
endptDIRSRVR     OBJECT-TYPE
                 SYNTAX      OCTET STRING
                 MAX-ACCESS  read-only
                 STATUS      current
                 DESCRIPTION "LDAP Server address. This variable returns
                    a 4-octet string specifying the IP address of the
                    LDAP server to be used for the phone's Directory
                    application."
                 ::= { endptAdvApps 2 }
endptDIRTOPDN    OBJECT-TYPE
                 SYNTAX      DisplayString
                 MAX-ACCESS  read-only
                 STATUS      current
                 DESCRIPTION "Directory Top Name. This variable returns
                    a string specifying the Topmost Distinguished Name in the
                    LDAP Directory to be used for the 4630's Directory
                    application."
                 ::= { endptAdvApps 3 }
endptDIRFULLNAME  OBJECT-TYPE
                 SYNTAX      DisplayString
                 MAX-ACCESS  read-only
                 STATUS      current
                 DESCRIPTION "Directory FullName Label. This variable returns
                    a string specifying the LDAP Directory's Full Name Field
                    Label, to be used for the 4630's Directory application."
                 ::= { endptAdvApps 4 }
```

```
endptDIRTELNUM  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Directory Phone Number Label. This variable returns
           a string specifying the LDAP Directory’s Telephone Number Field
           Label, to be used for the 4630’s Directory application."
        ::= { endptAdvApps 5 }
endptDIRSRCHTIME  OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Directory Search Duration. This variable returns
           an integer specifying the maximum number of seconds the LDAP
           Directory should be searched before reporting completion or
           failure of the search."
        ::= { endptAdvApps 6 }
endptDIRSRVRPWD   OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Directory Access Password. This variable returns
           a string specifying the password used for accessing the LDAP
           Directory.  The null strong is returned if no password has
           been specified."
        ::= { endptAdvApps 7 }
```

```
endptDIRUSERID   OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Directory Access User ID. This variable returns
           a string specifying the User identity used for accessing the LDAP
           Directory.  The null strong is returned if no user ID has
           been specified."
        ::= { endptAdvApps 8 }
endptDIRCODING  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Directory Coding Scheme. This variable returns
           a string specifying the character coding scheme (e.g. ASCII
           or Latin-1) to be used for LDAP Directory queries."
        ::= { endptAdvApps 9 }
endptDIRSTAT    OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Directory Connection Status. This variable returns
           an integer specifying the result code for the most recent
           Bind Request sent by the 4630 to the LDAP server."
        ::= { endptAdvApps 10 }
```

```
endptFTPSRVR    OBJECT-TYPE
                SYNTAX      OCTET STRING
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "FTP Server address. This variable returns
                   a 4-octet string specifying the IP address of the
                   FTP server to be used for the 4630’s file storage and
                   retrieval."
                ::= { endptAdvApps 11 }
endptFTPDIR     OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "FTP Server directory. This variable returns
                   a path name of the directory on the FTP server
                   to be used for the 4630’s file storage and retrieval."
                ::= { endptAdvApps 12 }
endptPHNEMERGNUM  OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Emergency number. This variable returns
                   a string specifying the emergency number to be outpulsed
                   when the user presses the optional Emergency button in
                   the 4630 Speed Dial application."
                ::= { endptAdvApps 13 }
```

```
endptPHNNUMOFCA   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Number of Call Appearances. This variable returns
           the number of Call Appearances downloaded from the DEFINITY
           during registration to the 4630 and 4620."
        ::= { endptAdvApps 14 }
endptPHNNUMOFFB   OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Number of Feature Buttons. This variable returns
           the number of administered Feature Buttons downloaded from
           the DEFINITY during registration to the 4630 and 4620."
        ::= { endptAdvApps 15 }
endptWEBCODING  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Web Coding Scheme. This variable returns
           a string specifying the character coding scheme (e.g. ASCII
           or Latin-1) to be used for characters entered on forms on
           web pages. Used by the 4630 Web application."
        ::= { endptAdvApps 16 }
```

```
endptWEBEXCEPT  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Web Proxy Server Exception List. This variable returns
           a string specifying a list of one or more HTTP proxy server
           Exception domains, separated by commas with no intervening
           spaces, up to 127 total characters.  Used by the 4630 Web
           application."
        ::= { endptAdvApps 17 }
endptWEBHOME    OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Web application URL. This variable returns
           a URL of the Home page for the 4630's Web application."
        ::= { endptAdvApps 18 }
endptWEBPORT    OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "Web application TCP port. This variable returns
           an integer specifying the TCP port most recently used for
           the HTTP proxy server.  Used by the 4630 Web application."
        ::= { endptAdvApps 19 }
```

```
endptWEBPROXY   OBJECT-TYPE
            SYNTAX      OCTET STRING
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION "Web Proxy Server. This variable returns
               an IP addresses, in dotted-decimal or DNS format, of an
               HTTP proxy server.  Used by the 4630 Web application."
            ::= { endptAdvApps 20 }
endptDIRLDAPPORT OBJECT-TYPE
            SYNTAX      INTEGER
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION "LDAP Directory Port. This variable returns
               the port number for LDAP messages, and has a default value
               of 389."
            ::= { endptAdvApps 21 }
endptVMLCODING OBJECT-TYPE
             SYNTAX      DisplayString
            MAX-ACCESS  read-only
            STATUS      current
            DESCRIPTION "Voice Mail Coding Scheme. This variable returns
               a string specifying the character coding scheme (e.g. ASCII
               or Latin-1) to be used for characters entered on forms on
               the 4630 Voice Mail application."
            ::= { endptAdvApps 22 }
```

```
endptVMLHOME    OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Voice Mail application URL. This variable returns
                   a URL of the Home page for the 4630 Voice Mail application."
                ::= { endptAdvApps 23 }
endptCLACTIVE    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Call Log activation flag. This variable returns
                   0 if the call log status is off/disabled, else 1."
                ::= { endptAdvApps 24 }
endptWMLCODING  OBJECT-TYPE
                SYNTAX      DisplayString
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "4620 Web Coding Scheme. This variable returns
                   a string specifying the character coding scheme (e.g. ASCII
                   or Latin-1) to be used for characters entered on forms on
                   web pages. Used by the 4620 Web application."
                ::= { endptAdvApps 25 }
```

```
endptWMLEXCEPT  OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "4620 Web Proxy Server Exception List. This variable returns
          a string specifying a list of one or more HTTP proxy server
          Exception domains, separated by commas with no intervening
          spaces, up to 127 total characters.  Used by the 4620 Web
          application."
        ::= { endptAdvApps 26 }
endptWMLHOME    OBJECT-TYPE
        SYNTAX      DisplayString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "4620 Web application URL. This variable returns
          a URL of the Home page for the 4620's Web application."
        ::= { endptAdvApps 27 }
endptWMLPORT    OBJECT-TYPE
        SYNTAX      INTEGER
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION "4620 Web application TCP port. This variable returns
          an integer specifying the TCP port most recently used for
          the HTTP proxy server.  Used by the 4620 Web application."
        ::= { endptAdvApps 28 }
```

```
endptWMLPROXY   OBJECT-TYPE
                SYNTAX      OCTET STRING
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "4620 Web Proxy Server. This variable returns
                   an IP addresses, in dotted-decimal or DNS format, of an
                   HTTP proxy server.  Used by the 4620 Web application."
                ::= { endptAdvApps 29 }
endptCTISTAT    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "CTI flag. This variable returns
                   0 if the CTI application is set to off, else 1."
                ::= { endptAdvApps 30 }
endptCTIUDPPORT OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "CTI application UDP port. This variable returns
                   an integer specifying the UDP port most recently used for
                   the UDP listener to receive broadcast messages."
                ::= { endptAdvApps 31 }
```

```
endptSTKSTAT    OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "Stock Ticker flag. This variable returns
                   0 if the 4630 Stock Ticker application is set to off, else 1."
                ::= { endptAdvApps 32 }

endptAdjuncts   OBJECT IDENTIFIER ::= { endpointMIB 6 }
endptFKEU       OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "FKEU flag. This variable returns
                   0 if the Feature Key Expansion Unit is off, disabled,
                   or not present, else 1."
                ::= { endptAdjuncts 1}
endptFKEUHEALTH OBJECT-TYPE
                SYNTAX      INTEGER
                MAX-ACCESS  read-only
                STATUS      current
                DESCRIPTION "FKEU flag. This variable returns
                   1 if the most recent self-test of the Feature Key Expansion
                   Unit or the 4620 failed, else 0."
                ::= { endptAdjuncts 2}
END
```

Appendix B: Creating Websites for the 4630 IP Telephone

B

Introduction

This Appendix describes the capabilities and limitations of the web browser in the 4630 IP Telephone and provides suggestions for designing websites for viewing on the 4630. It is intended for 4630 IP Telephone Web Browser [web page] designers, and assumes readers are familiar with HTML, Style Sheets, and ECMAScript.

This Appendix serves three primary functions. It:

- Presents the technologies that have been implemented in the 4630 IP Telephone Web Access application, including any limitations or non-standard implementations,
- Provides a suggested model for developing effective web pages to be viewed in the browser, and
- Provides sample HTML, Style Sheets, and ECMAScript that demonstrate the suggested model.

This Appendix is not intended to provide technical details on setting up a web server, nor does it provide information on web server technologies. Finally, this document is not intended to provide an introduction to web browser protocols or technologies.

NOTE A link to sites where HTML templates are available for customizing can be found on the Avaya website.

General Background

The 4630 IP Telephone display is a quarter VGA (320 pixels wide by 240 pixels high, 256 colors supported) display.

The data types and other features supported in the browser include:

- HTML 4.0
- Cascading Style Sheets (CSS), Levels 1 and 2
- Document Object Model (DOM) Level 1
- Images: GIF and JPEG
- ECMAScript (JavaScript) 1.4
- HTTP 1.1

- Secure Sockets Layer (SSL) 3.0
- Cookies

Browser Features and Behavior

This section presents the technologies that have been implemented in the 4630 IP Telephone Web Access application, along with any limitations or non-standard implementations. Since style sheets have become the preferred mechanism for controlling web page appearance and have made obsolete most attribute specifications with tags, the majority of attributes for tags were not tried. The attributes that were tested were those of real interest to specific tags. The browser is HTML 4.0 compliant, with a few minor differences as noted below.

NOTE In the sub-sections that follow, comments specific to the 4630 IP Telephone and its browser are shown in italics.

Document Skeleton

Certain tags define the basic framework of an HTML document. While most browsers are normally good at dealing with missing tags, when style sheets are applied it is essential that the tag structures be followed. Even ignoring style sheets, it is considered good style to follow the HTML rules. The following tags make up the basic skeleton of an HTML document:

- <html> indicates the start of an HTML document.
- <head> indicates the start of an HTML document's header. Title, meta definitions, ECMAScript function definition, document level style sheet definition, and external style sheet inclusion are all done in the header section.
- <title> sets the title of the document. Normally this is shown in the frame of the browser window. *On the phone, the title is shown in the Top Line Information Display Area.*
- <body> indicates the start of the body of an HTML document. The rest of the document will be embedded between the start and end <body> tags.
- <meta> is used to add additional information about an HTML page. *This is typically used by web walking tools, is of little use in browsers, and has no effect on the phone.*

Content-Based Style

Content-based tags deal with identifying words, phrases or chunks of text, or images as having a specific meaning or context. The intent of content-based tags is not to show physically marked text. However, it is not wrong for a browser to show content-based text in a distinct style, such as font style.

Each content-based tag is shown with a brief description, and any physical effects imposed by the browser.

- <abbr> indicates an abbreviation. *Using this tag has no effect.*
- <address> indicates an address. *Using this tag has no effect.*
- <acronym> indicates an acronym. *The enclosed text is shown in an italic font.*
- <cite> indicates a bibliographic citation or reference. *The enclosed text is shown in an italic font.*
- <code> indicates source code of a program. *The enclosed text is shown in a monospaced font.*
- <del> indicates deleted text. It is intended to show editorial markup. *The enclosed text is shown with a line through it.*
- <dfn> indicates a definition for a term. This is normally used with the first appearance of a term in a document. *The enclosed text is shown in an italic font.*
- indicates emphasis on a string of text. This is the contextual equivalent of using the <i> (italic) tag. *The enclosed text is shown in an italic font.*
- <ins> indicates inserted text. It is intended to show editorial markup. *The enclosed text is shown underlined.*
- <kbd> indicates keyboard input. This is similar to the <code> tag, but with a different contextual intent. *The enclosed text is shown in a monospaced font.*
- <samp> indicates literal text. *The enclosed text is shown in a monospaced font.*
- indicates emphasis on some text, but in a stronger contextual sense than the tag. This is the contextual equivalent of using the <b> (bold) tag. *The enclosed text is shown in a bold font.*
- <var> indicates source code variables. This is similar to the <code> tag, but with a different contextual intent. *The enclosed text is shown in an italic font.*

Logical Style

Tags that control logical style provide a similar functionality to those that control content-based style, but they have no implicit visual characteristics or contextual meaning. Instead, they only provide a means of invoking a style sheet. These tags, like all others, may have an ID and class attribute set. Using style sheet rules for these class names and IDs allows the designer to provide decoration, font, and color styles to each section. Thus, these tags may be considered to be designer-defined content-based tags:

- indicates a division. A newline is inserted between the previous text and the text following the tag. No physical changes are noticeable, except those implemented in style sheets.
- indicates a spanned section of text is placed immediately after the text preceding the tag with a newline. No physical changes are noticeable, except those implemented in style sheets.

Physical Style

Physical tags are effectively the opposite of content-based tags. The text in a physical tag may have no meaning whatsoever, outside of what the designer intended. These tags show text in a distinct style. Physical tags typically affect font style.

Each physical style tag is shown below with a brief description, and any physical effects imposed by the browser.

- <b> indicates that the text should appear in a bold font weight.
- <big> indicates that the text should appear one point size larger than the current text.
- <i> indicates that the text should appear in an italic font.
- <small> indicates that the text should appear one point size smaller than the current text.
- <sub> indicates that the text should appear as a subscript to the current text. The text is shown one point size smaller.
- <sup> indicates that the text should appear as a superscript to the current text. The text is shown one point size smaller.
- <tt> indicates that the text should appear as teletype text. The text is shown in a monospaced typeface font.

Physical Spacing and Layout

Physical spacing and layout tags define the basic structure of a document. Headings, paragraphs, tables and lists all provide some basic physical and logical layout to a page.

Each spacing and layout tag is shown with a brief description, and any unusual behavior is described.

- <p> indicates the start of a new paragraph. A blank line is inserted between previous text and text following the <p> tag.
-
 indicates that a breakpoint should be inserted. A newline is inserted between previous text and text following the <br> tag.
- <pre> indicates that the following text should have no formatting rules applied to it. This implies that no wrapping will be applied to this text, which may result in a horizontal scrollbar being added to view the text.
- <hr> indicates that a newline and a horizontal rule (line) should be inserted between the previous text and text following the <hr> tag.
- <blockquote> indicates that the following text is a quote, and should be offset in some way. The embedded text is shown with newlines before and after the text, and spacing to the left and right. Within that "block" of text, normal wrapping rules are applied.
- <q> indicates that the following text is a short quote, and should be shown in double-quotes (according to documentation). This text appears as normal text, with no physical identifying indications.
- <h1> - <h6> indicate heading sections. Each level is one point smaller than the previous level, with <h1> being the largest.

The section titled Maintaining Context contains design guidelines for maintaining user context.

Lists and Tables

For presentation of data in a logical format, lists and tables are two of the more important sets of tags. Since space is at a premium in the browser, tables should be used with discretion. While they provide sets of text, which are easy for the eye to process, they can quickly become cumbersome if misused in this browser. Both width and height are limited, so the data provided in the table cells should be short pieces of text. Beyond that, the page designer should come up with a different model for presenting the data, possibly using lists and divisions.

Lists

- starts an ordered list. This provides a list with some type of numbering: upper-case letters, lower-case letters, digits, upper-case roman numerals, lower-case roman numerals. Setting the type attribute is ignored in the browser.
- starts an unordered list. This provides a list with bullets to the left of each item.
- adds an item to an ordered or unordered list.
- <dl> starts a definition list. This is a list with two fields per list item. The first is a term and the second is the term's definition.

- <dt> adds a new definition term to a definition list.
- <dd> adds a new term definition to a definition list.

Tables

The new HTML 4.0 table tags, <thead>, <tbody> and <tfoot> are all implemented. These tags are intended primarily to allow printed pages to have headers and footers appear on each page, when the table is longer than a single page. Since the 4630 IP Telephone Web Browser does not provide the ability to print, it is recommended that these tags not be used.

- <table> starts a table layout.
- <caption> adds a caption to a table. The align attribute allows a caption to be placed above or below the table. This attribute has no effect in the phone.
- <tr> adds a new table row to a table.
- <th> adds a header for some column(s) of a table.
- <td> adds a piece of data for some row(s) and column(s) of a table.
- <thead> defines a set of table header rows. The intent of this tag is to provide a set of header rows for each printed page. While viewing the page in a browser, the header has no more effect than the <th> tag. What it does provide for the designer is a more logical breakout of the data. It is easy to recognize the header area of the table since it is set between the <thead> start and end tags.
- <tbody> defines the main body of a table, when used in conjunction with the <thead> and <tfoot> tags.
- <tfoot> defines a footer for a table. This tag may contain multiple rows. Like the <thead> tag, the intent of this tag is to provide a set of footer rows for each printed page. Viewing the page in a browser has no additional effect. And like the <thead> tag, this tag provides a logical breakout of the data for the designer.
- <colgroup> defines a column group. It may be used as a single definition of identical columns or as a container for dissimilar columns. The span attribute defines what columns are parts of each group.
- <col> controls the appearance of one or more columns within a column group.

Images

The 4630 IP Telephone Web Browser handles both GIF and JPEG image formats. No other image formats, included animated GIFs, are supported. Note that images take up a large amount of memory (compared to text) and updating the display can be an issue; therefore, it is recommended that images be kept to a minimum.

The image tags that may be used on the browser are as follows:

- <img> displays an image.
- <map> display a client-side image. Client-side images have shaped regions called areas. These areas are tied to URLs or ECMAScript functions. When an area in a region is clicked, the ECMAScript function is executed.
- <area> defines an area in a <map> image. Each area uses the href attribute to define a URL to jump to, or some ECMAScript to execute, when the mouse is clicked on that area.

See the Design Guidelines section for information on displaying images.

Links

Hyperlinks are the heart of the web browser's power. The link is what allows the user to click on some text or an image to jump to another web site, another page within this site, or another area in this page. Although URLs allow various protocols to be used, only http and https should be used for the 4630 IP Telephone Web Browser.

- <a> specifies the full or relative URL for a hyperlink. When using the target attribute, the _blank name should never be used. With the exception of frames, the 4630 IP Telephone Web Browser is not intended to be a multi-window browser. Thus, if a page attempts to reference a new, non-frame window, the browser will become confused and a reboot will be necessary.
- <base> defines the default target to be used in all <a> tags. This is done by setting the target attribute of the <base> tag. If the target attribute has been set in the <a> tag, the <base> tag value is ignored.

The use of the target attribute with the Web Access Application is strongly discouraged because the application supports a single window for browsing. The use of the target attribute may cause the browser software to create a new window that is outside of the Web Access Application's control.

Frames

Frames allow multiple windows to be created on the browser's base window. The browser is effectively split up into multiple areas. Each frame may be given a name or ID. Using the target attribute of the <a> tag, a web page may be displayed in another frame.

Although frames behave as defined, the small size of the screen makes them impractical to use. Just like tables, frames take up too much room on the display. One could, though, take advantage of frames by having a small "table of contents" frame with images for links. In general, however, use of frames is strongly discouraged.

- <frameset> defines an area for a set of frames.
- <frame> defines a single frame in a frameset.
- <iframe> defines an in-line frame. More detail
- <noframes> provides a fallback for browsers that don't handle frames. If the designer intends to use frames and make these pages available to phone, this tag should certainly be employed.

See the Design Guidelines section for information on displaying frames.

Forms

HTML forms provide the user the ability to enter data into a web browser. This data can then be passed on the web server for processing. It is difficult to predict what web designers might have in mind for the 4630 IP Telephone Web Browser. However, all forms-related tags have been verified, and the results are presented here.

- <form> defines the basic input form, and defines the action to be performed when the submit button is selected, via the action attribute.
- <input> defines most user input. The type attribute defines the type of input to be used. The <button> tag was created to replace type values of button, reset and submit, in a cleaner, more flexible way. Other type values available are checkbox, hidden, image, password, radio, and text. The type value file is useless in the context of this browser, since the intent of this control is to allow the user to select a file on their local disk.
- <button> defines a button that the user may select. This behaves much like the <input type=button> tag, except that the physical appearance is three-dimensional and any text, image, or combination, may be displayed in the button.
- <fieldset> encapsulates a section of a form's contents to create a group of related form controls. The phone's browser puts a simple box around the fieldset.
- <label> associates a relationship between a form control and one or more text labels. Labels may be tied to form controls by the for attribute in the label and the id attribute in the form control. They may also be tied by embedding the form control inside the <label> tag (for example, <label>Name: <input type=text id=name></label>).
- ·<legend> gives a label to a <fieldset> tag. This label appears at the top of the fieldset section of the form, with a line separating the legend from the rest of the fieldset.
- <optgroup> provides nested, cascading menus to the user. This doesn't seem to work.

- <option> defines the values available in a <select> scrolling list or drop-down menu.
- <select> defines scrolling lists and drop-down menus.
- <textarea> provides free-form user input and display. This provides a scrolled text area for the user to read or type text.

Character Entities

As with any syntactic language, HTML has certain characters that have special meaning. The two most obvious of these characters are the < and > symbols, which surround all tags. These characters cannot be typed in directly if the designer's intent is to display these characters. Thus, all characters that can be displayed in a web browser have numeric values assigned to them. In addition, many of these characters have names also assigned. The numeric values are entered into the source web page as &#nnn; where nnn is some 3 digit value. For example, the < symbol is entered as '<'. Name values are entered into the source web page as &name; where name is the name associated with this character. Again, using the < symbol, this would be entered as '<'. The set of characters defined by the World Wide Web Consortium are fully supported in the browser in conformance with the standard.

Colors

The browser supports 256 colors. Colors may be specified by name, RGB percentages or RGB raw numbers. The HTML and CSS specifications suggest 16 named colors. The 4630 IP Telephone browser recognizes these color names, which are: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow. Beyond these 16 well-known names, it is recommended that RGB percentages or raw numbers be used for specifying colors.

Fonts

Font specifications are one of the most important styles that may be applied to a web browser. Because of the size of the screen on the 4630 IP Telephone, the browser has only a single font available for use. Font weights, such as normal and bold, are supported, although finer values, such as lighter and bolder are not. Normal and Italic font styles are also supported. Font sizes are also supported, specified by either percentages or raw numbers can be used. However, percentages below 50% all appear as the same size.

Although some problems were found with font specifications, given the size and resolution of the screen, fonts behave reasonably well. The only major problem found is the inability to specify font families.

See the Design Guidelines section for information on displaying images.

Cookies

Cookies can be a useful feature in maintaining the state of a user when interacting with a web site. HTTP provides no state information, such as when or how often a user has visited a site. Cookies allow web sites to track this information by storing a simple set of values on the browser for the current session. Normally, browsers also provide the ability to save cookies to disk, so this information may be retained between sessions. However, the 4630 IP Telephone browser has no such ability to save any data between sessions. Cookies do behave well within the realm of the current session, and may be employed if desired. The 4630 IP Telephone has a maximum of 8Kilobytes of data available for cookie storage.

Design Guidelines

This section presents guidelines for developing a good model that effectively presents web pages to be viewed on the 4630 IP Telephone Browser. The biggest challenge in designing pages for this browser is the limited amount of space available for viewing the pages. The 4630 IP Telephone screen is a ¼ VGA display. Part of that screen is lost to the browser by the main controls of the 4630 IP Telephone. Thus, page layout must be done effectively and efficiently so as to avoid causing more space to be lost through additional screen controls such as scrollbars. Font sizing will make or break the usability of a page. A balance must be found between fitting as much text as is possible, and allowing users to read the text without straining their eyes.

Fixed-Width Objects

One of the primary functions of a web browser is to present text, wrapped at the right (or left when dealing with internationalization) border of the browser window. The browser always attempts to avoid adding a horizontal scrollbar. However, if fixed-width objects, such as tables and pre-formatted (<pre> tags) text are used, the browser makes wrapping a secondary priority to presenting the data exactly as the HTML dictated.

Web browsers do not resize themselves larger when scrollbars are added. Thus, if there are more lines of text than can fit in the height of the browser window, a vertical scrollbar is added. This now takes up some of the width of the browser, and less text is displayed on a single line. If text is wider than the width of the browser, a horizontal scrollbar is added. This now takes up some of the height of the browser, and fewer lines of text are displayed.

Given the small size of the browser, it is highly desirable to avoid causing scrollbars to appear. Unless the amount of text shown in a page is kept to a bare minimum, it is unlikely that a vertical scrollbar will be avoided. However, avoiding fixed-width objects, or ensuring that the size of fixed-width is kept small, will gain some viewable space on the browser.

In general, if scrolling is a requirement, vertical scrolling is better perceived by users than horizontal scrolling.

Images

The use of images in a web page is always a concern. For example, a page with many images can cause downloading to be slow. While this is still a concern in the phone, the size of an image has a much greater effect. Memory in the browser, and in the phone in general, is limited. Each image will use a sizable amount of memory, and the browser may become overwhelmed. An image should only be used if it is essential to a page.

Images also fall into the realm of fixed-width objects. All images should be checked to verify that they don't cause a horizontal scrollbar to be added. An image may be scaled down by the browser, by setting the width and height attributes of the <img> tag. The designer may instead choose to scale the images when setting up the web site. This avoids forcing the browser to deal with the sizing (using the width and height attributes scales the image after it is downloaded by the browser) and speeds up downloading of the images. Finally, reducing the size of the image reduces the amount of memory used.

Animated GIF images are a bad idea since they use up quite a bit of memory. Additionally, because of the persistence of the LCD screen, animated images tend to smear in the browser and lose their effectiveness. Hence, animated GIFs are considered "not supported" with this phone.

Frames

While frames provide a useful method of browsing a series of pages (via a contents frame and a document text frame), they also use up real estate, just like scrollbars. Even if the frame decorations were all turned off, the frame containing the majority of the document text now suffers the problems discussed above; the width of the frame is smaller, and the chances of adding a horizontal scrollbar have increased. Additionally, a single line of text will have fewer characters viewable, and the page becomes even more difficult to read and comprehend. Using simple navigation buttons at the top and bottom of the page, or even at the top and bottom of each section should make up for the missing contents frame.

The interaction between frames and scrollbars is another important area of concern. While most browsers manage scrolling within each frame independently, the 4630 Web Access Application only scrolls the entire (single) window. This moves the user's view of the frames as a whole, but it does not scroll any of the data in individual frames. There will likely be data in frames that can never be seen.

The combined problems of minimal screen real estate and the scrolling issues really makes frames something simply to avoid.

Fonts

Font size is a major concern in the browser. Without designer intervention, the browser displays text as if it were running on a PC in a normal width and height display. This means that the fonts shown would appear huge relative to the screen size. As a result, only seven lines of text will be viewed in the browser when no font sizing is applied. Additionally, only about 25 characters would be viewed per line. This would generally be unusable from a practical point of view.

To make the browser usable, some form of font sizing should be performed to allow a reasonable amount of text to be viewed on one screen. Ideally, font sizing should be done in a single external style sheet, and all pages should reference this style sheet. Setting font sizes in document-level style sheets, or even worse, in in-line style sheets or <font> tags, makes it very difficult for the designer to update font changes, and runs the risk of failing to make the change everywhere. By using a single external style sheet, consistency in the pages is managed.

Defining fonts may be done in a static external style sheet or by dynamically generating a style sheet using some form of server-side application, such as a CGI script or a Java servlet. Using dynamic generation of a style sheet allows the server to decide, per user, what font size to use. This in turn allows the user to notify the server to change the font size. All HTML pages would have an external style sheet reference, which is another server-side executable. This executable would dynamically generate font information based on the cookie, form component or configuration value, relative to the IP address of the phone.

Using a static external style sheet makes page development and testing much easier, since how it appears to the designer would be how it appears to the reader. Allowing the user to choose a font size forces more work on the designer to verify that all fixed size items appear acceptably

Maintaining Context

Given the small working area, it is easy for the user to become lost. Headings will often not be in view, and the user, if distracted by other work, can lose their sense of context. Style sheets may be used to help maintain this sense of context through color. The and tags are intended to provide page designer-defined content-based style. By defining classes of tags with various colors and border styles, the designer may provide additional information. It is important to bear in mind that a significant number of people suffer from some form of color blindness. Thus, it may be necessary to design pages both with and without using color styles.

User Interaction

As discussed above, HTML forms work reasonably well in the browser. However, due to the limitations of the phone as an input device, keyboard input can be difficult. A complete keyboard is made available based on the context of mouse selection. When a text input control is on-screen, the user simply needs to click on the input control. The keyboard appears, with the browser thrown into a small scrolling area. The input control is roughly centered in the scrolling area. The user may then press the software keys and the text is shown in the input control. Simply pressing the done button dismisses the keyboard, and the input control shows the newly typed text. While this interaction technically works fine, from a user perspective, it can be difficult to type a large amount of text. Thus, unless user input is absolutely necessary, it should be avoided. When necessary, user input should be kept to a minimum.

Appendix C: Creating Websites for the 4620 IP Telephone

C

Introduction

This Appendix describes the capabilities and limitations of the web browser in the 4620 IP Telephone and provides suggestions for designing websites for viewing on the 4620. It is intended for 4620 IP Telephone Web Browser (web page) designers, and assumes readers are somewhat familiar with WML.

This Appendix serves the following functions:

- Presents the portions of WML that have been implemented in the 4620 IP Telephone Web Access application, including any limitations or non-standard implementations
- Provides considerations for developing effective web pages to be viewed in the browser

This Appendix is not intended to provide technical details on setting up a web server, nor does it provide information on web server technologies. Finally, this document is not intended to provide an introduction to web browser protocols or technologies.

General Background

The 4620 Telephone has a 168 pixel-by-132 pixel four-gray scale LCD display. The area of the display available for presenting a WML web page to the user is 168 pixels across by 96 pixels in height, arranged in 6 rows each 16 pixels in height. In addition, the top row will display the web page title (if any) and the bottom row presents up to four softkey labels at one time, each a maximum of 6 characters, that are used for <do> tags.

The data types and other features supported in this browser include:

- WML 1.2
- HTTP 1.1

A chart at the end of this Appendix summarizes the more detailed information that follows.

NOTE Unsupported WML 1.2 tags will not be rendered and will not cause the browser to fail. Unknown tags and misspelled tags will cause an error message.

WML Document Skeleton

Certain tags define the basic framework of a WML document. The tags listed below make up the basic skeleton of a WML document, and are supported by the 4620 unless otherwise indicated.

- Common tag attributes: xml:lang, class, and id

The attributes xml:lang, class and id are universal attributes associated with every WML element. The web browser will support these tags as follows:

Attribute	Comments
xml:lang	NOT SUPPORTED
class	NOT SUPPORTED
id	SUPPORTED

- <**wml**> tag - The <wml> tag defines a deck of cards and encloses all information the deck and is a required WML element. This tag must contain at least one <card> tag.
- <**head**> tag - The <head> tag is an optional WML tag containing information relating to the deck as a whole, including meta-data and access control elements. This tag is not supported.
- <**meta/**> tag - The optional <meta> tag is contained between multiple <head> tags. This tag gives values for the parameters that describe the content of the deck. This tag is not supported.
- <**card**> tag - A single WML file can contain multiple cards. This leads to the analogy - a "deck" of "cards" within a single WML file. A "card" is essentially the specification of one specific WML page. This is a mandatory tag.

The card element attributes supported by the web browser are as follows (unsupported attributes are indicated as such in the Comments column):

Attribute	Value(s)	Description	Comments
newcontext	true false	Re-initializes the browser context Default is "false"	Clears out the current WML browser context. This entails emptying the navigation stack history and clearing out all variables. NOT SUPPORTED

Attribute	Value(s)	Description	Comments
ordered	true false	Specifies the order of card content. When ordered is set to "true" the browser will display the content in a fixed order. When ordered is set to "false" the users will decide the order as they navigate between content. Default is "true"	Optional; sets a Boolean value that provides information on how the content of the current card is arranged. Used by the browser to organize the display presentation and layout. If set to true, content is organized in a linear sequence of elements (for example, a series of ordered or non-optional input elements). If set to false, content is in no natural order (for example, a series of unordered or optional input elements). The default is true. NOT SUPPORTED.
title	*cdata*	The title of the card	Can be used for title displays. SUPPORTED
onenterbackward	*url*	Occurs when the user navigates into a card using a “prev” task	SUPPORTED
onenterforward	*url*	Occurs when the user navigates into a card using a “go” task	SUPPORTED
ontimer	*url*	Occurs when a “timer” expires	SUPPORTED

- <**template**> tag - The <template> tag defines a template for all the cards in a deck. The "code" in the <template> tag is added to each card in the deck. Only one <template> tag for each deck may be specified. This tag can only contain <do> and <onevent> tags.

The **template** tag attributes supported by the 4620 web browser are as follows:

Attribute	Value(s)	Description	Comments
onenterbackward	*url*	Occurs when the user navigates into a card using a “prev” task	SUPPORTED
onenterforward	*url*	Occurs when the user navigates into a card using a “go” task	SUPPORTED
ontimer	*url*	Occurs when the “timer” expires	SUPPORTED

NOTE The implication for rendering WML pages is that the local environment will always override a global template for <do> types with the same name and type. If there is a onevent in the template and a *local* onevent of the same type, the local onevent takes precedence over the global one.

- <**access**> - The <access> tag limits access within the deck to certain cards. This tag is not supported.

Text Elements

- <**br**> tag - The <br> tag tells the browser to add a line break to the text at the point the element is written.
- <**p**> tag - The <p> tag specifies a paragraph of text with alignment and line wrapping properties. All text data must be contained inside this tag. Only <do> tags, wml and card elements can exist outside of the <p> tag. When rendered, this tag will cause subsequent text to begin on the next line.

Attribute	Value	Description	Comments
align	left right center	Aligns the paragraph. Default is "left"	SUPPORTED
mode	wrap nonwrap	Sets whether a paragraph should wrap lines or not	Since horizontal scrolling is not supported, all text lines are wrapped. WML pages with mode=nowrap are ignored, and the text will wrap. NOT SUPPORTED.

The following tags are not supported but the content inside the tags will be rendered as normal text:

- <**table**> tag - The <table> tag specifies a table. This tag is not supported.
- <**td**> tag - The <td> tag defines individual cell contents in each row of a defined table. This tag is not supported.
- <**tr**> tag - The <tr> tag defines each row of a defined table. This tag is not supported.

Text Formatting Tags

The following tags are not supported but the content inside the tags will be rendered as normal text:

- <**b**> tag - The <b> tag specifies bold text. This tag is not supported.
- <**big**> tag - The <big> tag specifies large font text. This tag is not supported.
- <**em**> tag - The tag specifies emphasized text. This tag is not supported.
- <**i**> tag - The <i> tag specifies italicized text. This tag is not supported.
- <**small**> tag - The <small> tag specifies text using a small font size. This tag is not supported.
- <**strong**> tag - The tag specifies strongly emphasized text. This tag is not supported.
- <**u**> tag - The <u> tag specifies underlined text. For the 4620 Web Access application, only links appear underlined. This tag is not supported.

Anchor Elements

- <**a**> tag - <a> elements define <go> tasks that require a URL link specification. All <a> tags will be rendered as underlined. All <a> nested tags (br, img) are supported. One anchors may be rendered is rendered per line. The user can select the link by pressing the left Feature Buttons associated with that display line.

Attribute	Value	Description	Comments
href	*url*	REQUIRED. Defines where to go when the user selects the link	SUPPORTED
title	*cdata*	Defines a text identifying the link	SUPPORTED
accesskey			NOT SUPPORTED

- <**anchor**> tag - <anchor> elements define <go> tasks that require a URL link specification. All anchors are rendered as underlined. All <anchor> nested tags (br, go, img, prev, and refresh) are supported. A maximum of 6 anchors may be rendered on the screen at one time. The user selects a link by using the Left Feature button associated with that display line.

Attribute	Value	Description	Comments
title	*cdata*	Defines a text identifying the link	SUPPORTED

Image Elements

- <**img**> tag - The <img> tag is used to place an image in the text flow. The monochrome wbmp (wireless bitmap) format is used to code images for a wireless display, but the 4620 browser instead renders the alternative text supplied with the image, as specified by the alt attribute.

Attribute	Value	Description	Comments
align	top middle bottom	Aligns the image	NOT SUPPORTED
alt	*cdata*	REQUIRED. Sets an alternate text to be displayed if the image is not displayed. If this is not supplied, either default text is displayed (if available) or the following message is displayed: "Image not displayed".	SUPPORTED
height	*px (pixel)* %	Sets the height of the image	NOT SUPPORTED
hspace	*px* %	Sets white space to the left and right of the image	NOT SUPPORTED
localsrc	*cdata*	Sets an alternate representation for the image	NOT SUPPORTED
src	*url*	REQUIRED if image is to be displayed; not used if text is displayed instead	NOT SUPPORTED
vspace	*px* %	Sets white space above and below the image	NOT SUPPORTED
width	*px* %	Sets the width of the image	NOT SUPPORTED

Event Elements

- <**do**> tag - The <do> tag is a card-level user interface. It serves as a general mechanism for the user to activate a task, typically performed by the user clicking on a word or phrase in the display. A task is performed in response to an event. There are four tasks in WML: go, noop, prev, and refresh.

The mandatory *type* attribute provides information about the intent of the element, helping to improve processing. If the 4620 web browser does not recognize the specified type, (e.g., testing, experimental, and vendor specific types), the specified type is treated as unknown. The browser only renders WML 1.2 tags. Any other tags will cause an error and the user will receive an error statement "not a valid wml page".

Attribute	Value	Description	Comments
type	accept prev help reset options delete unknown x-* vnd.*	REQUIRED. Defines the type of the “do” element	SUPPORTED
label	*cdata*	Creates a label for the “do” element	Optional; creates a string label for the element. The 4620 browser imposes a six character limit. SUPPORTED
name	*mmtoken*	Defines a name for the “do” element	SUPPORTED
optional	true false	If set to true, the browser ignores this element. If set to false, the browser does not ignore this element. Default is “false”	Optional. SUPPORTED

Type	Description	Comments
accept	Acknowledgement of acceptance	SUPPORTED
delete	Delete item	SUPPORTED
help	Request for help	SUPPORTED
options	Options or additional operations	SUPPORTED

Type	Description	Comments
prev	Backward navigation	SUPPORTED
reset	Clearing or reset	SUPPORTED
X-*n or x-*n	Experimental	SUPPORTED, but treated as 'unknown'

<**do**> tags are rendered as softkey labels on the bottom line of the 4620 display. <do> tags are specified per WML page and therefore are page context-sensitive. The eight "do" types are labeled either specifically in a WML page or by a browser-dependent label.

If no labels are given, then the "do" types will have the following default labels:

Type	Default Label if no label specified
accept	SELECT
delete	DELETE
help	HELP
options	OPTIONS
prev	BACK
reset	RELOAD
X-*n or x-*n	UNKNOWN
Vnd* Any mix of upper or lower cases	AVAYA (Available for future use, but currently UNKNOWN)

If no <do> tags have been specified, no softkeys will be displayed:

If one <do> tag is specified the following softkeys will be displayed:

1st DO			

If multiple <do> tags are specified, display them as follows:

1st DO	2nd DO	3rd DO	MORE

Page 1 Softkeys:

1st DO	2nd DO	3rd DO	MORE

Page 2 Softkeys:

4th DO	5th DO	Etc.	MORE

NOTE If more than one page of softkey labels are specified, pressing the MORE softkey automatically presents the user with the next page of labels. If the last page is displayed and the user presses the MORE softkey, the first page of labels is then displayed. As implied by the above, the softkey buttons are labeled in sequential order of the <do> tags.

- <**onevent**> tag - The onevent tag serves as a container for code that you wish to be executed automatically when one of the four intrinsic events occurs. The onevent element is said to bind (associate) the tasks (code) to the event for the element. The intrinsic event must be specified using the mandatory type attribute.

For example, when a user presses the BACK softkey, instead of being routed to the previous screen, the user will be directed to another specified page because this tag carries out a onevent backward event.

The intrinsic events are:

Event	Permitted Tags	Description
onenterbackward	card or template	Occurs when a <prev> navigates back onto a card. SUPPORTED.
onenterforward	card or template	Occurs when a <go> navigates into a card. SUPPORTED
onpick	option	Occurs when an item is selected/unselected by a user. SUPPORTED
ontimer	card or template	Occurs when the time expires. SUPPORTED

The template element creates code that is inserted into all cards in a single deck. The nested tags are as follows, go, noop, prev, and refresh.

There are no visual implications for supporting the <onevent> tag.

Attribute	Value	Description	Comments
type	onenterbackward onenterforward onpick ontimer	REQUIRED. Specifies the type of the “onevent” element	SUPPORTED

- <**postfield**> tag - The postfield tag is used to set a name/value pair that can be transmitted to an origin server (source of the request) during a URL request. The name is set by the name attribute and must be a valid WML variable name. The value is set by the value attribute. There are no visual rendering implications with this tag.

Attribute	Value	Description	Comments
name	*cdata*	REQUIRED. The name of the field	SUPPORTED
value	*cdata*	REQUIRED. The value of the field	SUPPORTED

Task Elements

- <**go**> tag - The go element can contain one or more postfield elements. If the destination of a go element is a card within the same deck, all of the postfield elements will be ignored. The go element can also contain one or more setvar elements. Unlike postfield elements, there are no destination limitations on passing information contained in the setvar elements. The <go> nested tags (postfield and setvar) are supported.

Attribute	Value	Description	Comments
href	url	REQUIRED	SUPPORTED
accept-charset	Charset_list	A comma- or space-separated list of character encoding the server must be able to process. The default value is "unknown".	SUPPORTED
method	post get	Sets how to send the data to the server. Default method is get. When method="get", the data is sent as a request with*?data* appended to the URL. The disadvantage of get is that it can be used only for a limited amount of data; if you send sensitive information it will be displayed on the screen and saved in the web server's logs.With method="post", the data is sent as a request with the data sent in the body of the request. This method has no limit, and sensitive information is not visible.	SUPPORTED

Attribute	Value	Description	Comments
sendreferer	true false	If set to true, the browser sends the URL of the current deck with the request, which allow servers to perform simple access control on decks, based on which decks are linking to them. Default is "false"	SUPPORTED

- <**noop**> tag - The noop tag dictates that no operation should be done. This tag can be used on the card level to prevent an event, specified on the deck level by the template element, from occurring. It can only be contained in either a do or onevent element.

An example of noop is to use a <do> tag to add a "Back" link to the card. When users click on the "Back" link, generally they should be taken back to the previous card. However, the <noop> tag prevents this operation; when the user clicks on the "Back" link nothing happens.

- <**prev**> tag - The prev tag specifies navigation to the previous URL in the history.
- <**refresh**> tag - The refresh tag specifies a refresh task whereby the whatever card is being displayed will be refreshed. This task specifies the need for an update of the user agent context as specified by the contained <setvar> elements.This tag can only be nested inside an anchor, do, or onevent element. Xml:lang is not an associated attribute. User-visible side effects of the update can occur during the processing of the <refresh>.

Input Elements

- <**input**> tag supported - The input tag specifies a point where the user is prompted to enter text.

Attribute	Value	Description	Comments
name	*nmtoken*	REQUIRED. The name of the variable that is set with the result of the user's input	SUPPORTED
emptyok	true false	Sets whether the user can leave the input field blank or not. Default is "false"	SUPPORTED

Attribute	Value	Description	Comments
format	 A a N X x M m *f nf	Sets the data format for the input field. Default is "M" A = uppercase alphabetic or punctuation characters a = lowercase alphabetic or punctuation characters N = numeric characters X = uppercase characters x = lowercase characters M = all characters m = all characters *f = Any number of characters. Replace the f with one of the letters above to specify what characters the user may enter. nf = Replace the n with a number from 1 to 9 to specify the number of characters the user may enter. Replace the f with one of the letters above to specify what characters the user can enter. The user cannot exit the input box unless the correct number or type of characters is entered. The user does not receive an error message if incorrect data is entered	SUPPORTED
maxlength	*number*	Sets the maximum number of characters the user can enter in the field	SUPPORTED
size	*number_of_char*	Sets the width of the input field	NOT SUPPORTED
tabindex	*number*	Sets the tabbing position for the input field	NOT SUPPORTED
title	*cdata*	Sets a title for the input field	SUPPORTED
type	text password	Indicates the type of the input field. The default value is "text"	SUPPORTED
value	*cdata*	Sets the default value of the variable in the "name" attribute	SUPPORTED

The six display lines of the 4620 associated with feature buttons are all available for input elements. The top line of the display cannot be used for input.

The input tag causes an automatic line break before and after input text.

Only 1 input tag can exist per display line.

When a user views a page with the input tag specified, the first thing that shows up in the top line is the card title, if specified. When the user scrolls to the first line containing input, the top line shows the input box title if specified, otherwise the card title is shown. The top line displays the card title for all non-input text.

When the input box is selected, a vertical line (the "cursor") appears at the left side of the input box.

The attribute type password should only be used when it is important to not display the user' s password on the screen. Asterisks will be displayed instead. It is also important that the password not be cached.

The phrase [enter text here] appears for all input tags.

Only the correct size, type, and number of characters are accepted in to the input box. For example, if alpha text is specified and the user types in a symbol or numeric text, the user input is not accepted. The screen repaints and the user has to re-enter their input. If the wrong kind of text is typed, the user receives an error tone. If the "n" (number) value is specified and the user types in the incorrect number of characters, their input will not be accepted.

- <**fieldset**> tag - The fieldset tag is used to group logically related elements in a card. This tag is not supported.
- <**optgroup**> tag - Sets of <optgroup> brackets can be put around <options> in a <select> list. The results in breaking a list into sublists.

Attribute	Value	Description	Comments
title	*cdata*	Sets a title for the optgroup element	SUPPORTED

- <**option**> tag - A set of option tags is needed to specify each individual item in a list. This tag must be used with the select tag.

Attribute	Value	Description	Comments
onpick	*url*	Sets what is going to happen when an user selects an item	SUPPORTED
title	*cdata*	Sets a title for the option	SUPPORTED
value	*cdata*	Sets the value to be used when setting the "name" variable in the select element	SUPPORTED

- <**select**> tag - The select tag allows for the definition of a list, embedded in a card, to allow the user to choose inputs from a list rather than having to type something in. The select tag must be used with the option tag.

Attribute	Value	Description	Comments
name	*nmtoken*	Names the variable that is set with the index result of the selection	SUPPORTED
ivalue	*cdata*	Sets the pre-selected option element. If none is specified the first item in a list is automatically selected.	SUPPORTED
multiple	true false	Sets whether multiple items can be selected. Default is "false". False is used for a single selection.	SUPPORTED
tabindex	*number*	Sets the tabbing position for the select element	NOT SUPPORTED
title	*cdata*	Sets a title for the list	SUPPORTED
value	*cdata*	Sets the default value of the variable in the "name" attribute	SUPPORTED

A single select is rendered with a small square containing a dot. A multiple select is rendered as multiple squares - blank if there is nothing in them, else a lower case "x".

Variable Elements

- <**setvar**> tag - There are no visual rendering implications with this tag.

Attribute	Value	Description	Comments
name	*cdata*	REQUIRED. Sets the name of the variable	SUPPORTED
value	*cdata*	REQUIRED. Sets the value of the variable	SUPPORTED

- <**timer**> tag - The timer tag sets a timer that starts counting. This tag must be used with <onevent type="ontimer"> to be useful.

Attribute	Value	Description	Comments
value	*cdata*	REQUIRED. Sets the default value of the variable defined in the "name" attribute	SUPPORTED
name	*nmtoken*	REQUIRED. Names the variable that is set with the value of the timer	SUPPORTED

Character Entities

As with any syntactic language, WML has certain characters that have special meaning. The two most obvious of these characters are the < and > symbols, which surround all tags. These characters cannot be typed in directly if the designer's intent is to display these characters. Thus, all characters that can be displayed in a web browser have numeric values assigned to them. The numeric values are entered into the source web page as &#nnn; where nnn is a three-digit value. For example, the < symbol is entered as '<'.

In addition, many of these characters also have names assigned. Name values are entered into the source web page as &name; where name is the WML name associated with this character. For example, the < symbol would be entered as '<'. The set of characters defined by the World Wide Web Consortium are fully supported in the 4620 browser in conformance with the standard.

For convenience, a few of these key symbols are specified below.

Description	Symbol	Numeric Entity	Name Entity
double quotation	"	"	"
ampersand	&	&	&
apostrophe	'	'	'
less than	<	<	

Colors and Fonts

The browser supports a 4-grayscale display. Because of the size of the screen on the 4620 IP Telephone, the browser has only a single font available for use, which is based on Latin-1. Only a normal font weight is supported. Bold, italic and different font sizes are not supported. The font used by the 4620 defines characters to have at most six pixels in width.

Summary Of WML Tags And Attributes

Tag	Attribute	Supported?
<a>		Yes
	accesskey	No
	href	Yes
	title	Yes
<access>		No
<anchor>		Yes
	title	Yes
<b>		No
<big>		No
 		Yes
<card>		Yes
	newcontext	No
	onenterbackward	Yes
	onenterforward	Yes
	ontimer	Yes
	ordered	No
	title	Yes
<do>		Yes
	label	Yes
	name	Yes
	optional	Yes
	type	Yes, except x-*
		No
<fieldset>		No

Tag	Attribute	Supported?
<go>		Yes
	accept-charset	Yes
	href	Yes
	method	Yes
	sendreferer	Yes
<head>		No
<i>		No
<img>		Yes
	align	No
	alt	Yes
	height	No
	hspace	No
	localsrc	No
	src	No
	vspace	No
	width	No
<input>		Yes
	emptyok	Yes
	format	Yes
	maxlength	Yes
	name	Yes
	size	No
	tabindex	No
	title	Yes
	type	Yes
	value	Yes
<meta>		No
<noop>		Yes
<onevent>		Yes
<optgroup>		Yes
	title	Yes
<option>		No
	onpick	Yes
	title	Yes
	value	Yes
<p>		Yes
	align	Yes
	mode	No
<postfield>		Yes
	name	Yes
	value	Yes
<prev>		Yes

Tag	Attribute	Supported?
<refresh>		Yes
<select>	 *ivalue* *multiple* *name* *tabindex* *title* *value*	Yes Yes Yes Yes No Yes Yes
<setvar>	 *name* *value*	Yes Yes Yes
<small>		No
		No
<table>		No
<td>		No
<template>	 *onenterbackward* *onenterforward* *ontimer*	Yes Yes Yes Yes
<timer>	 *name* *value*	Yes Yes Yes
<tr>		No
<u>		No
<wml>		Yes
{Universal Attributes}	*xml:lang* *class* *id*	No No Yes